電腦時代

COMPUTING AGE 17

工商業 • 教育 • 家庭娛樂 • 私人應用電腦雜誌　港幣十元

高質素產品
大衆化價錢

PARA SYSTEMS

5¼" mini floppy disk

日本製造，保証100%無誤差

AUTHORIZED AGENT
SOFT GARDEN
深水埗福華街黃金商場地庫16B
3-7289066

MD2D MINI-FLOPPY DISK

資宝牌 軟磁盤

* High quality mini-floppy disk for data storage and retrieval.
* 100% error-free of disk surface ensured by comprehensive tests.
* Prolongs drive read/write heads by its superior surface smoothness.
* Best selection of jacket optimize drive characteristic.

Item Model No.	MD1D	MD2D
Disk Size (Inch)	5¼	5¼
Sector	Soft	Soft
Usable Side	Single	Double
Density	Double	Double
Tracks per Inch (TPI)	48	48
Number of Tracks	40	80
Storage Capacity (Bytes)	163,840	327,680

Super 5

A & N ENTERPRISE CO., LTD.
20/F WESTERN CENTRE,
40-50 DES VOEUX ROAD, WEST, H K
TEL: 5-461179 5-488463
TELEX: 65218 MCLWW HX
FAX: 5-497149

Super 5
DOT MATRIX PRINTER
EN-P1091/EN-P1090
点阵式打印机
NEW VERSION
EP－1201
EN-P1091
• Print Mode Selector Switch
• Near-letter quality printing
• Print Speed 120 cps.
EN-P1090
• High resolution, dot-addressable graphics
• Print speed 80 cps.
100% IBM* PC
MPATIBLE
SUPER 5 MBE-420 15" PRINTER • Print speed 180 cps.
SUPER 5 EP-1807 15" 7 COLOR PRINTER
• Inked platens printer
NEW
SUPER 5 CP80/CPA80 9 pin printer
SUPER 5 MONITOR FOR IBM PC 资料显示器
SUPER 5 POWER SUPPLY 变压器
for IBM* pc Max. output Power 133.9W
SUPER 5 5.25" Floppy Diskette – EE50(SSDD)/EN-320K (DSDD)
SUPER T-40e/C-40 DISK DRIVE 软盘磁碟驱动器
Plus FDD for IBM* PC
CHINON 502 / EPSON SP-521 / TEAC FD-55B / MATSUSHITA JA551-2
* IBM PC is a registered trademark of International Business Machine Corp.
Super 5
EI-EN ELECTRONICS, LTD.
22-23-213 Ebisu-I-Chome Shibuya-Ku, Tokyo, Japan.
A & N ENTERPRISE CO., LTD.
20/F., Western Centre, 40-50 Des Voeux Rd., West, H.K.
Tel: 5-461179 (3 lines) 5-488463 Telex: 65218 MCLWW HX.

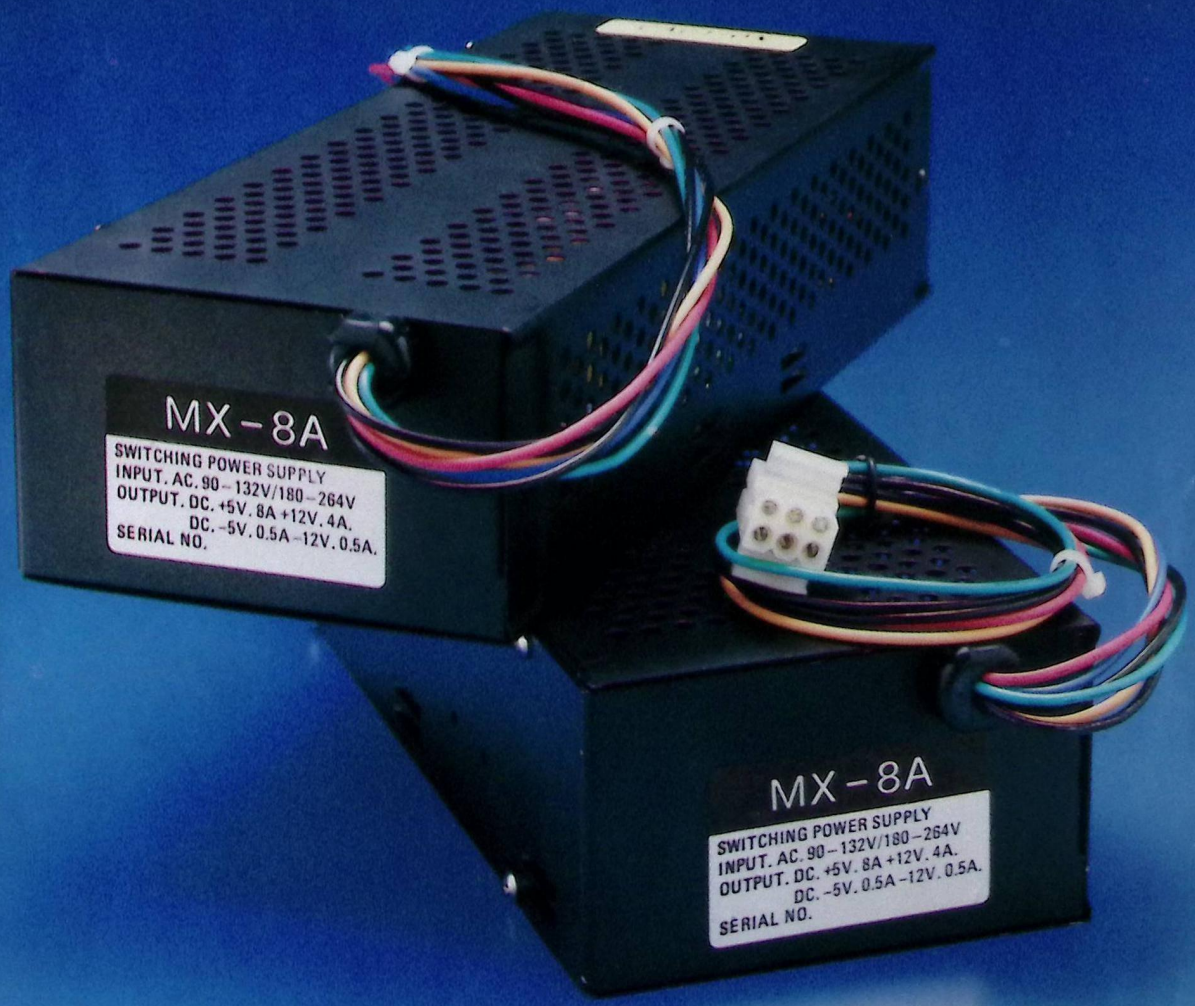
MX-8A 電源供

MX-8A

功能十足・令主機如

10小時動態測試・短路保護試驗・一年免費保

歡迎出口或批發
敬希垂詢



適用於AP及各種單板機，最高負載如下：

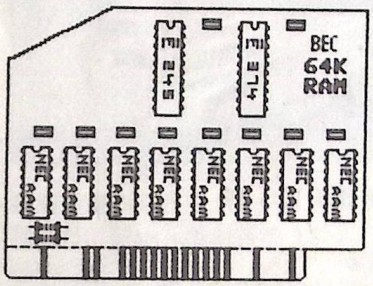
INPUT AC・90-132V/180-264V
OUTPUT DC・+5V・8A+12V・4A
DC −5V・0.5A−12V・0.5A

香港電工器材
電話： 3-876673

LEE'S ELECTRONIC CO.

李氏電子公司

H.K. $130



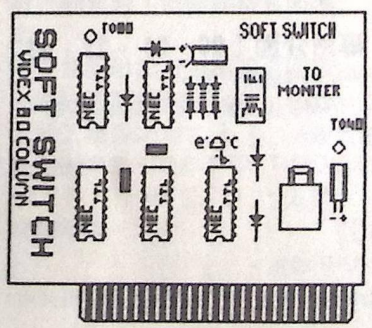
大量現貨供應

APPLEWORKS SAYS 55K

80 COLUMN + EXTEND 64K = BEC 64K RAM

- BEC 64K RAM PLUS INTO YOUR APPLE //e TO MAKE A 128K SYSTEM
- BEC 64K RAM OFFERS TOTAL COMPATIBILITY
- BEC 64K RAM OFFERS DOUBLE RESOLUTION CAPABILITY (560 X 192)
- BEC 64K RAM CAN BE USED WHILE PROGRAMMING UNDER APPLE COMPUTER'S PRO DOS

H.K. $90



WHAT IS A SUCCESSFUL 80-COLUMN SOFTSWITCH ?

- ACCEPT GR, HGR, AND HGR2 COMMANDS
- Gbasic AND Dgraph CAN DISPLAY GRAPHIC
- PFS FILE AND PFS REPORT CAN REMAIN ON 40-COLUMN DISPLAY
- PFS GRAPHIC CAN REMAIN ON 40-COLUMN DISPLAY AND DISPLAY VARIOUS CHARTS
- DOLLAR $ENCE CAN BE DISPLAY ON 80 COLUMN AS WELL AS VARIOUS CHARTS

ONLY OUR SOFTSWITCH CAN DO ALL ABOVE !

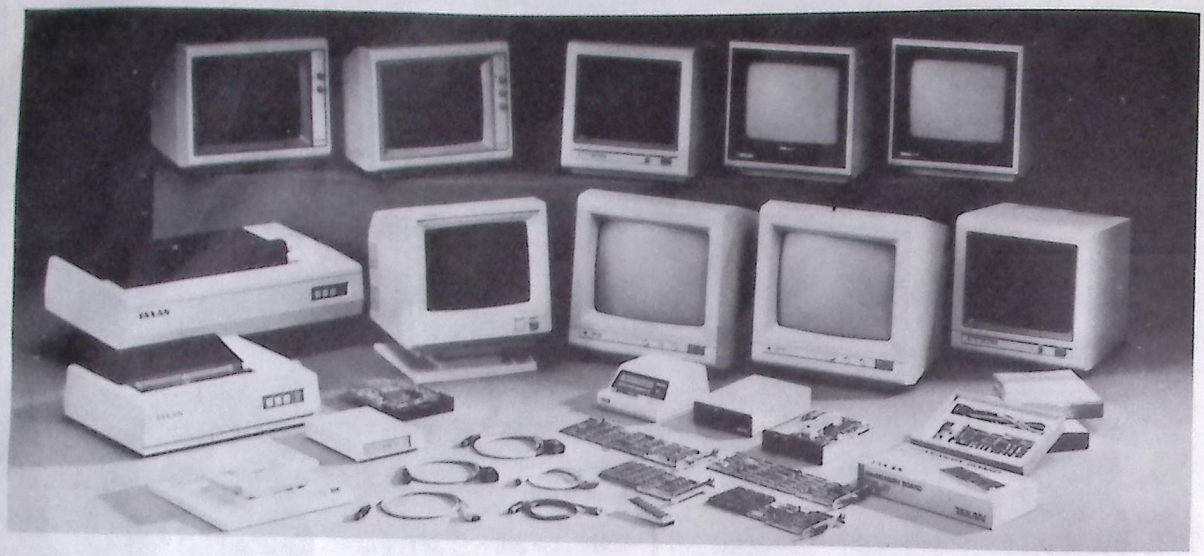
九龍 深水埗 黃金商場
地庫 66A 室
電話:3-7291022

Basement Shop NO.66A,
Golden Bldg. Shopping Centre,
146-152, Fuk Wah Street,
Sham Shui Po, Kowloon.
Tel. : 3-7291022

香港 北角 英皇道
國都商場 三樓 213 室
電話:5-718392

213 Olympia Shopping Centre,
255 King's Road,
North Point,
Hong Kong.
Tel. : 5-718392

TAXAN 泰山電腦產品



單色電腦螢幕（1，2）

- 不反光、方形、平面、深底色螢幕設計，解像力特高，視頻寬達20MHz，畫面更清晰。
- KX 1201 - （P31綠色）深綠底色顯像管，適合一般電腦。
- KX 1212 - （P39綠色）深灰底色長殘影顯像管，適合IBM PC/XT* 及同類型產品。

「RGB」彩色電腦螢幕（3，4，5，8，11）

- 能顯示任何顏色組合，率先採用「線性影像放大系統」。
- 絕不干擾磁碟及其他電器。
- 適用於各款電腦。

「混合訊號」彩色電腦螢幕（9，10）

- 獨特功能，可隨意選擇綠色或琥珀色（橙色）顯示。
- 內置音响系統，2W輸出功率，適合一般錄影機及多類型電腦使用。
- PAL或NTSC兩種系統可供選擇。

點矩陣式印字機（6，7）

- 印字速度每秒140字。
- 高質素字體(NEAR LETTER QUALITY)。
- 墨帶壽命特長。
- 可用普通紙及10吋或17吋電腦紙。
- 適用於IBM PC/XT* 及一般電腦使用。

螢幕底座（12）

- 盛托電腦螢幕，隨意調校方向及角度，符合人類工程學要求，更加方便舒適。

印字機記憶緩衝器（13，19）

- 64K 記憶系統。
- 可另增加記憶至256K。
- 不佔用電腦時間，尤其在印中文字及圖案時。
- 增加工作效率。

電腦螢幕接線（14，15，16，17，18）

- 各款螢幕接駁線，適合不同電腦使用。

IBM PC/XT* 專用介面（20，21，22，23）

- KIF3210 - 256K RAM記憶擴展介面。
- KIF3210S - 256K RAM記憶擴展及串行式介面接口。
- KIF3600 - 彩色/圖案顯示介面。
- KIF3700 - 單色顯示及CENTRONICS平行式印字機介面接口。

電視調諧器（24）

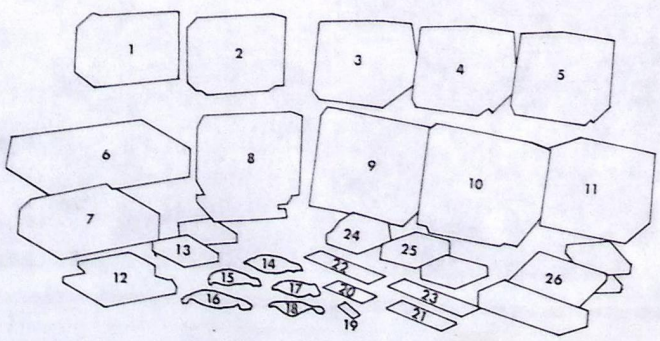
- PAL-D彩色系統，UHF線路。
- 接駁PAL系統電腦螢幕及適當天線，即可作電視機使用。

5.25吋磁碟機（25）

- 適用於APPLE II**或IBM PC/XT* 及同類型電腦。

APPLE IIe**電腦專用介面（26）

- 80字畫面及RGB彩色顯示介面。



總代理：
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司（請認明有限公司，並無分號）
九龍尖沙咀柯士甸路22-26號好進年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

* IBM PC / XT is registered trademark of International Business Machine Corp.
** Apple II & IIe are registered trademark of Apple Computer Inc.

TAXAN®

泰山超高解像度彩色電腦螢幕

SUPER VISION III

- 獨特三種文字顯示功能，綠色、琥珀色（橙色）及反白色。
- 高解像度 640 點乘 262 行，每點直徑0.37mm，視頻寬達 18 MHz 。
- 先進科技，採用SUPER-HIGH-CONTRAST 顯像管，畫面更清晰，久看不倦。
- 採用高品質零件，尤其在顯示彩色圖案時，仍能保持畫面穩定。
- 可另加配附設有石英鐘之底座，調節不同方向、角度，符合人體工程學之要求。
- 完全適用於 IBM PC*、PC/XT*, APPLE II** 及同類型電腦。

總代理：

VINCENT COMPUTER CENTRE LTD.

雲生電腦有限公司（請認明有限公司，並無分號）

九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

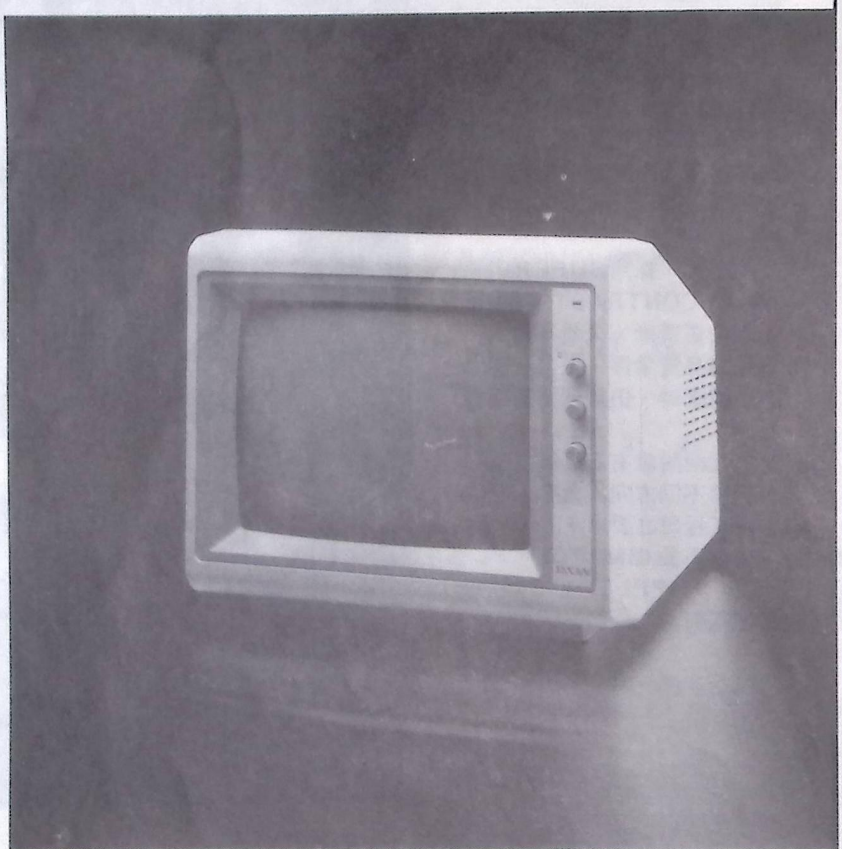
* IBM PC、PC/XT is registered trademark of International Business Machine Corp.

** Apple II is a registered trademark of Apple Computer Inc.

TAXAN®

泰山12吋綠色/琥珀色（橙色）電腦螢幕

KX-1212/1213



- 不反光、方型、平面、深灰底色（P39）長殘影顯像管。
- 視頻寬達20MHz以上。
- 解像力特高，畫面更清晰、穩定。
- 顯示字數2000字（80×25行）。
- 適合IBM PC* 、PC/XT* 及同類型產品。
- 全日本製造。

總代理：

VINCENT COMPUTER CENTRE LTD.

雲生電腦有限公司（請認明有限公司，並無分號）

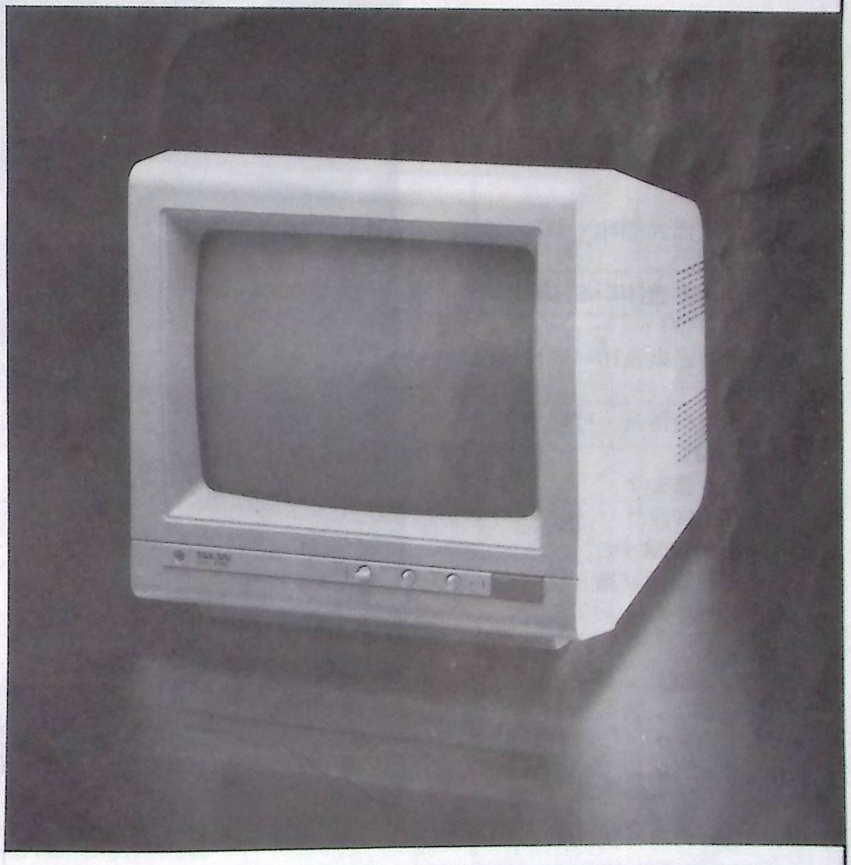
九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

* IBM PC、PC/XT are registered trademark of International Business Machine Corp.

TAXAN®

泰山14吋「混合訊號」彩色電腦螢幕

VISION PAL 220 COLOUR MONITOR



- PAL系統（VISION PAL）及NTSC系統（220 COLOR MONITOR）可供選擇。
- 獨特功能，可隨意選擇綠色或琥珀色（橙色）螢幕。
- 內置音响系統，2W輸出功率，適合一般錄影機及多類型電腦。
- 全日本製造。

總代理：

VINCENT COMPUTER CENTRE LTD.

雲生電腦有限公司 （請認明有限公司，並無分號）

九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

TAXAN®

泰山點矩陣式打印機
KP-810/KP-910

- 高速來回式編印，每秒可印140字。
- 高質素字體(NEAR LETTER QUALITY)。
- 可用普通紙及10吋或17吋電腦紙。
- 墨帶壽命特長，可印字多達3百萬個。
- 可印9國文字。
- 超低噪音設計。
- 適用於 IBM PC* 、PC/XT* 及一般電腦使用。

總代理：
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司 （請認明有限公司，並無分號）
九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

*IBM PC, PC/XT are registered trademark of International Business Machine Corp.

★ 學習尖端科技
★ 邁向成功坦途

現代電腦系統訓練中心

MODERN COMPUTER SYSTEM TRAINING CENTRE

——各項電腦課程招生——

COURSE 課程表	上課編制	時數	學費
電腦語言課程			
(1) BASIC PROGRAMMING COURSE	個別或班制	16HRS	$280
(2) ADVANCED BASIC COURSE	個別或班制	15HRS	$300
(3) ASSEMBLER LANGUAGE COURSE	個別或班制	16HRS	$350
(4) "C" LANGUAGE COURSE	班制上課	16HRS	$800
(5) 兒童電腦班（LOGO及初級BASIC）	班制上課	10HRS	$200
電腦硬件課程			
(1) 6502電腦自嵌實習班（軟硬件理論兼備）	個別或班制	30HRS	$500
(2) APPLE-II電腦修理班	班制上課	80HRS	$1500
(3) IBM-PC電腦修理班	班制上課	80HRS	$1800
綜合應用課程			
(1) dBASE-II入門及商業應用初級及高級班	個別或班制	10HRS	$300
(2) CP/M操作系統	個別或班制	16HRS	$350
(3) DOS應用及拆碟班	班制上課	20HRS	$550
(4) WORDSTAR TRAINING COURSE	個別或班制	16HRS	$350
(5) PEACHTREE A/C SYSTEM會計系統	個別或班制	16HRS	$800

上述學費概以班制上課爲準，個別教授另議

創辦叁年　電腦碩士任教

本校師資　(1)香港及中文大學電腦系，英美加電腦系畢業生。　(2)英國曼徹斯特大學電腦碩士，英國電腦學會會員。

本中心開放時間：10：00A.M—10：00P.M

地址：九龍深水埗福華街156號黃金商場地庫20號室，54號室及67號室。
電話：3—601531　3—617206

香港教育司署立案（1957年創校）

世界函授學校

規模大　優點多　科目好

初級電腦科
電腦基本原理，電腦操作，邏輯電路，程式編寫。八個月畢業。

初級無綫電科
全部圖解教導，原理裝修並重，極易學習，十個月畢業。

半導體裝修科
配合進步的技術，具初學或職業的裝修人員必修的學科。十個月畢業。

實用數理
可以加強數理基礎，費用少，收效大，6個月畢業。

$x+y=z$

實用商科
著重實際應用，加強商業學識，期間短，學習易，六個月畢業。

無綫電工程
設計，研究和深造的理想學科。一年半畢業。

汽車工程
適合職業及業餘人員加強對汽車方面之學識。

冷氣工程
冷氣，冷藏人員缺乏，專門學校極少，一年半畢業。

商業美術科
內容新，範圍廣，包括美術和最新廣告設計。一年畢業。

應用漫畫
漫畫可訓練頭腦，增強思想，取材新，學習易，六個月畢業。

導師
國內外著名大學學士，經香港教育司考核合格，教導認眞，決不馬虎。

講義
全部經教育司署審定，閱讀容易，印刷精美，免費供應。

費用
除低廉之學費外，學生決無額外負擔。

文憑
本校頒發文憑，均經政府簽証，實是最佳學歷証件。

校址： 香港九龍彌敦道五六六號9樓A座
電話：③—803532

本港學生可利用電話詢問索取章程　電話③－803532

請依綫剪下，填好姓名，回郵地址，貼在信封上投寄，本校即免費寄上簡章

郵件請寄：P.O.BOX 70011, KOWLOON, CENTRAL POST OFFICE
九龍中央郵政局信箱 70011 號

技術就是財產

（寄自）
姓名：
地址：
（請註明要學的科目）
□初級電腦科
□無綫電工程科
□初級無綫電科
□半導體裝修科
□實用數理科
□汽車工程
□冷氣工程
□實用商科
□商業美術
□應用漫畫

請貼上郵票

香港教育司署立案

To：世界函授學校註册處
P. O. BOX No. 70011
KOWLOON CENTRAL POST OFFICE,
HONG KONG.

香港九龍中央郵局信箱70011號

電話：K-803532

學問就是力量

專業電腦
PROFESSIONAL COMPUTER

歷史悠久　　深受推崇

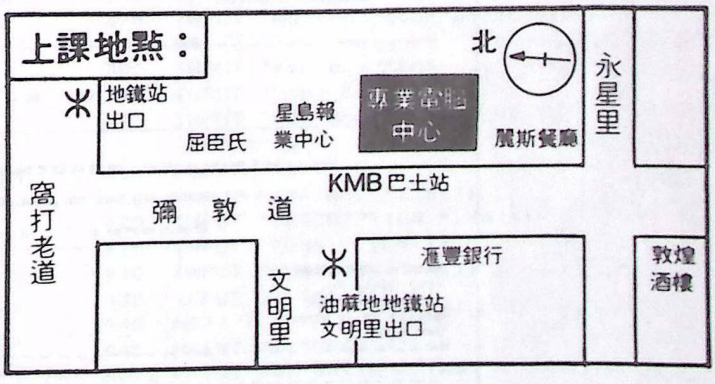
推廣電腦知識
贈閱半年電腦雜誌

親愛的學員：您們好。為推廣電腦知識，凡於1985年8月30日之前參加下列任何一項課程，各學員均享有電腦雜誌贈閱半年的優惠。價值港幣陸拾圓，讓大家在進修期間，有良好的參考讀物

課　程：

1. 基本電腦概念及程式設計
2. BASIC 程式設計
3. 中學會考電腦科補習
4. 6502電腦裝嵌、修理、職業先修課程
5. WORDSTAR 商業電腦打字課程
6. MULTIPLAN及VISICALC操作及應用
7. DBASE Ⅱ 操作及應用
8. CP/M操作系統研究
9. Z-80及6502組合語言
10. 電腦入行職業先修實習證書速成課程

- 理論與實習並重
- 課程深入淺出，適合各階層人仕
- 每人一套電腦實習
- 結業後獲頒證書一份
- 成績優異者由本中心推薦加入電腦公司工作
- 學員可自由選擇實習時間
- 在課程期間，學員可隨時回校免費實習
- 可代公司訓練僱員
- 本中心備有中學會考電腦科參考資料
- 代購優惠服務（電腦雜誌　電腦軟件　磁碟　印字機　電腦文具　電腦書籍　磁碟機　電腦附件及各種名廠電腦）



索取章程，查詢及報名處：

專業電腦中心　九龍彌敦道486號益南華厦3樓電梯2字　電話：3-328248　3-318960

或　寶發書業專門店　九龍深水埗欽州街94號高登商場一樓76A, B　電話：3-602326

工商業發貨／報關消單

維恩

本程式是一個非常實用的商業發貨單／報關消息，用家能將有關資料儲存及印出，使用時需要配合一部有GRAPH—ER 咭的印字機（其他印字機界面咭亦可，但要作小許修改）。

假如閣下是一間公司或工廠負責人，而又剛開始使用電腦化處理日常出貨收貨事項，對市面一般的商業軟件又未熟習使用，則本程式肯定會十分適合你。

印出來的發貨單抬頭公司可任意更改至合適爲止，只要閣下對BASIC 有一點認識即可。以下是有關要更改字串的行句：

行句 110 改動公司名稱

例如將"HING LEE TRADING COMPANY" 改爲"COMPUTING AGE PUBLISHER"（請比較表A及B）

行句 125 改動公司地址

例如將"26, G／F., SHEUNG SHA WAN RD., KOW—LOON, HONG KONG" 改爲"1104, TUNG CHUN COMM. BLDG., 438—444 SHANGHAI ST., KLN."

行句 140 是單據的類形，例如INVOICE 或MONTHLY STATEMENT 亦可。

行句 200至220（較後的行句360至380）是用作印出各細則，各位可隨意改到適合自己的公司使用，有關細節內容可參考程式列表。

行句 230則是印出貨品內容細則，如重量、數量、單價及總額。

行句 245可修改輸出的單位，例如將KG改爲GRAM，$改爲US $等。

總括來說，這個程式有相當程度的商業應用價值。程式是用BASIC需成，對於只有少許電腦知識的商廠負責人相信會很適合。

表A及B是兩家公司的發貨單比較，程式(1)是發貨單的主程式，程式(2)是作局部修改的行句，作過如述的改動後，可印出表B的效果。

大家可用下列指令將程式(1)存入磁碟

SAVE INVOICE MAKER

或在本期的程式磁碟中找到相同的FILE。■

表A

COMPUTING AGE PUBLISHER

1104,TUNG CHUN COMM. BLDG.,438-444 SHANGHAI ST.KLN

Messrs.COMPUHOUSE
Your Cont. No.
Invoice No.1234
Sailed on or about
per S/S

Hong Kong

REF No.	DESCRIPTION	Pkgs.	Quantity	Price	Amount
	PROGRAM DISKETTE		3000Kg	12	$36000
	MAGAZINE ISSUE#17		20000Kg	10	$200000

TOTAL=$236000

E. & O. E.

表B

HING LEE TRADING COMPANY

26, G/F.,SHEUNG SHA WAN RD.,KOWLOON, HONG KONG

General Invoice

Messrs.COMPUHOUSE
Your Cont. No.
Invoice No.1234
Sailed on or about
per S/S

Hong Kong

REF No.	DESCRIPTION	Pkgs.	Quantity	Price	Amount
	PROGRAM DISKETTE		3000Kg	12	$36000
	MAGAZINE ISSUE#17		20000Kg	10	$200000

TOTAL=$236000

E. & O. E.

工商業發貨／報關消單

程式(1)主程式

```
0  REM  ********************
1  REM  *  INVOICE MAKER  *
2  REM  * BY MA CHUN HING *
3  REM  ********************
5  TEXT : HOME
10 B$ =  CHR$ (4):C$ =  CHR$ (14)
     :D$ = ""
15 E$ = "------------------------
     ----------------"
20  HOME
25  VTAB 8: PRINT "YOU MAY CHOOSE
      FROM FOLLOWING FUNCTION"
26  PRINT : PRINT
30  PRINT "1 PRINT I.HEADING & TH
     E INFORMATION"
35  PRINT "2 PRINT HEADING & THE
     INFORMATION"
40  PRINT "3 PRINT THE HEADING ON
     LY"
45  PRINT "4 ENTER THE INFORMATIO
     N"
50  PRINT "5 EXIT THIS PROGRAM"
60  VTAB 20: HTAB 20: CALL  - 868
     : INPUT "WHICH:?";S
65  IF S < 1 OR S > 5 THEN 60
70  IF S = 5 THEN  END
75  FOR I = 1 TO 25: PRINT : NEXT

80  ON S GOTO 100,100,100,350
100  PRINT "PRINT THE HEADING"
105  PRINT D$: PR# 3: PR# 1
110  PRINT  TAB( 16);C$;"COMPUTIN
     G AGE PUBLISHER "
120  PRINT
125  PRINT  TAB( 13);"1104,TUNG C
     HUN COMM. BLDG.,438-444 SHAN
     GHAI ST.KLN"
130  IF S = 3 THEN  PRINT D$: PR#
     O: GOTO 10
135  IF S = 2 THEN 150
137  PRINT
140  PRINT  TAB( 23);C$;"General
       Invoice": PRINT  TAB( 23);
     C$;"-----------------": PRINT

150  PRINT D$: PR# O
154 B$ =  CHR$ (4):C$ =  CHR$ (14
     ):D$ = ""
155  INPUT "THE FILE NAME OR THE
     INVOICE NO.";A$
160  GOTO 550
195  PRINT D$: PR# 3: PR# 1
200  PRINT "Messrs.";F$
205  PRINT "Your Cont. No.";O$
210  PRINT "Invoice No.";A
215  PRINT "Sailed on or about ";
     G$
220  PRINT "per S/S ";H$; TAB( 58
     );"Hong Kong ";I$
225  PRINT C$;E$
230  PRINT "REF No.    DESCRIPTION
                Pkgs.  Qu
     antity    Price     Amount"
235  PRINT C$;E$
240  FOR I = 1 TO N
241 M$ =  STR$ (M(I)):R$ =  STR$
     (R(I))
245  PRINT J$(I); TAB( 9);K$(I); TAB(
     41);L$(I); SPC( 8 -  LEN (L$
     (I)));M(I);"Kg"; SPC( 10 -  LEN
     (M$));R(I); SPC( 12 -  LEN (
     R$));"$";M(I) * R(I)
250 T = M(I) * R(I)
251 L = L + T
252 L =  INT (L * 100 + 0.5) / 10
     0
255  NEXT I
256  PRINT : PRINT
257  PRINT C$;E$
260  PRINT  TAB( 65);"TOTAL=$";L
275  PRINT D$: PR# 0
280  INPUT "ARE YOU PRINT THE E.&
      O.E.(Y/N)";Q$
285  IF Q$ = "N" THEN 310
295  PRINT D$: PR# 3: PR# 1
300  PRINT  TAB( 30);C$;"E. & O.
     E."
305  PRINT D$: PR# 0
310  INPUT "any other copy!(Y/N)"
     ;Y$
315  IF Y$ = "N" THEN 5
320  IF Y$ = "Y" THEN 100
350  PRINT "ENTER THE INFORMATION
     "
355  INPUT "THE FILE NAME OR THE
     INVOICE NO.";A$
360  INPUT "Messrs.";F$
365  INPUT "Your Cont. No. ";O$
370  INPUT "Invoice No.";A
375  INPUT "Sailed on or about ";
     G$
380  INPUT "per S/S ";H$
390  INPUT "Hong Kong ";I$
395  INPUT "HOW MANY STATEMENT TO
      ENTER";N
400  FOR I = 1 TO N
410  INPUT "REF NO. ";J$(I)
420  INPUT "DESCRIPTION ";K$(I)
425  INPUT "Pkgs. ";L$(I)
430  INPUT "Quantity ";M(I)
435  INPUT "Price ";R(I)
440  NEXT I
445  PRINT B$;"OPEN A";A$
450  PRINT B$;"WRITE A";A$
455  PRINT F$
460  PRINT G$
461  PRINT H$
462  PRINT I$
463  PRINT O$
464  PRINT N
465  PRINT A
```

工商業發貨／報關消單

```
466  FOR I = 1 TO N
467  PRINT J$(I)
468  PRINT K$(I)
469  PRINT L$(I)
470  PRINT M(I)
471  PRINT R(I)
472  NEXT I
525  PRINT B$;"CLOSE A";A$
530  GOTO 5
550  PRINT B$;"OPEN A";A$
551  PRINT B$;"READ A";A$
552  INPUT F$
553  INPUT G$
554  INPUT H$
555  INPUT I$
556  INPUT O$
557  INPUT N
558  INPUT A
```

```
559  FOR I = 1 TO N
560  INPUT J$(I)
561  INPUT K$(I)
562  INPUT L$(I)
563  INPUT M(I)
564  INPUT R(I)
565  NEXT I
567  PRINT B$;"CLOSE A";A$
570  GOTO 195
```

程式(2)行句例子修改

```
110  PRINT  TAB( 16);C$;"HING LEE
      TRADING COMPANY"
125  PRINT  TAB( 13);"26, G/F.,SH
     EUNG SHA WAN RD.,KOWLOON, HO
     NG KONG"
```

解決DOS 3.3 SUB-DIRECTORY的方法

青海

本人是貴刊的訂戶，從貴刊處得到不少的有用程式，並把他們分類收集在多張碟上，但由於貴刊的程式往往一個主程式帶幾個副程式，以致在複製時不知何者是主程式，何者是副程式，而要把一叠叠的電腦時代月刊，拿出來翻查，感到非常之費時失事。故特意找方法，希望能把資料寫在磁碟目錄上。

現在介紹的方法就是更改DOS 3.3 $B203的值。

在DOS 3.3內，有一個起自$B1F9的副程式，是用來比較要用的檔案名稱和在目錄磁區的檔案名稱。在$B203，本來存在着$1E（10進制是30，即檔案名稱最多能容許的字元個數），此副程式會比較兩個檔案的所有字元是否一樣，（所有檔案名稱都有30個字元，如不夠者，都會被「空白」所取代）如果完全一樣，則返回，否則，拿下個目錄磁區的檔案名稱作比較。

由此可見，$B203所存的值實在是比較檔案名稱字元的個數，當它的值被更改時，例如$0F，則DOS 3.3只會比較前15個字元是否一樣，如果一樣就返回，不同就取下一個。於是實際有效的檔案名稱只有15個字元。其餘15個DOS會視而不見，在這情況下：

HELLO—DEMO　　CA13

和HELLO—DEMO　　CBA2

會視爲同一檔案，假如你磁碟內是前者，拿你打入「RUN HELLO—DEMO　　**********」都可以拿出

「HELLO—DEMO　　CA13」

註：　*代表任何字元包括空格。

用這種方法，就可以把每個檔案的尾15個位元存放有關的資料，例如是否副程式，程式來源等。對於主程式，並不需修改所叫的副程式名稱，因爲DOS只比較前三十個位元。

當然，閣下未必一定把$B203改爲$0F，也可以改爲其他值，視乎閣下需要而定。但不能過$1E，（十進制30）因爲DOS 3.3沒有大過30個位元的檔案。■

四大磁碟系統磁碟資料存放格式詳析

DOS 3.3, CP／M, PASCAL, PRODOS

第一篇：DOS 3·3 磁碟資料儲存格式

前言

DOS 3.3、CP／M、PASCAL及PRODOS 是目前蘋果電腦最常用的四種磁碟操作系統。每種操作系統都有著本身獨特的優點。雖然我們經常聽聞到這幾個名字，甚至每日都接觸到這些操作系統，但未必每個讀者都能清楚了解到這幾個磁碟操作系統彼此之間的分別，它們到底是怎樣將資料放入磁碟之上。

在以後數期的電腦時代，我們將會和各位深入剖釋這四種操作系統究竟是怎樣擺放資料入磁碟之上，與及怎樣格式化一張新磁碟。

ZAP 程式是追尋磁碟系統程序的最佳工具，因爲我們可以ＺＡＰ程式去了解和觀看磁碟上每一個扇區或磁軌內的數據資料。

例如儲存於一片磁碟上的DOS（磁碟系統），若其中的程式在儲錄過程中有些資料損壞或流失，則整個ＤＯＳ 檔案處理系統亦可能會處於崩潰的狀態。比方說，各位APPLE Ⅱ的磁碟用家，你們會否遇過電腦不能讀入磁碟上的HEL—LO 程式呢，這便是儲錄於磁碟上的ＤＯＳ 出現了資料損壞。更嚴重的是ＤＯＳ系統資料的流失，可能整個磁碟檔案系統便會完全崩潰，電腦便永遠沒法讀出儲錄於磁碟上的其餘資料檔案。

雖然，衆多電腦軟件公司也曾推出一些拯救磁碟檔案的應用軟件，但這却比不上以ＺＡＰ程式去直接修改和研究整個磁碟與檔案系統的處理與分析過程。

一如IBMPC 的電腦用戶，APPLE Ⅱ用家也同樣可擁有很多不同形式的磁碟系統，以最流行的DOS 3.3 CP／M、PASCAL，以至可驅動硬磁碟周邊裝置的PRODOS 系統。

雖然，APPLE Ⅱ電腦還擁有一些更早期的磁碟系統版本，但以上述的４種最爲時尚和流行。它們是分別具有不同的特點和檔案處理功能，但基本上它們已趨於標準化和相容性的設計。因爲在不同的磁碟系統下，已有很多應用程式是可讀寫其他的磁碟系統中的資料檔案。

以下的文章中，我們將會把上述的４種ＤＯＳ系統的格式和檔案處理過程，逐一向各讀者來一個清楚的交代。

現在，請各位先參看圖１，它是針對４個不同磁碟系統的檔案儲存方法和共通的格式。以最外圍的PRODOS 緊接向內的PASCAL、CP／M和DOS 3.3 ，其磁碟扇面（DIS—K SECTOR） 的安排是一樣，但扇面編碼却各有不同。還有，圓中心的實物扇面（PHYSICAL SECTOR）是４種ＤＯＳ 的主要共通點，因爲PHYSICAL SECTOR 是由每種ＤＯＳ的扇面０開始以反時針方向，將整隻磁碟分成（$０－$Ｆ）16處扇面，故又稱之爲PHYSICAL SECTOR。這又可理解成當一片完全沒有資料的磁碟，一旦被格式的同時，其第零個扇面區的位置是隨機的，若被置定後，電腦便可將其餘的磁碟扇區再分爲15（一共16個PHYSICAL SEC—TOR）個扇面，而至於扇面是否順序編碼安排，這便要參看圖１了。

邏輯的編排

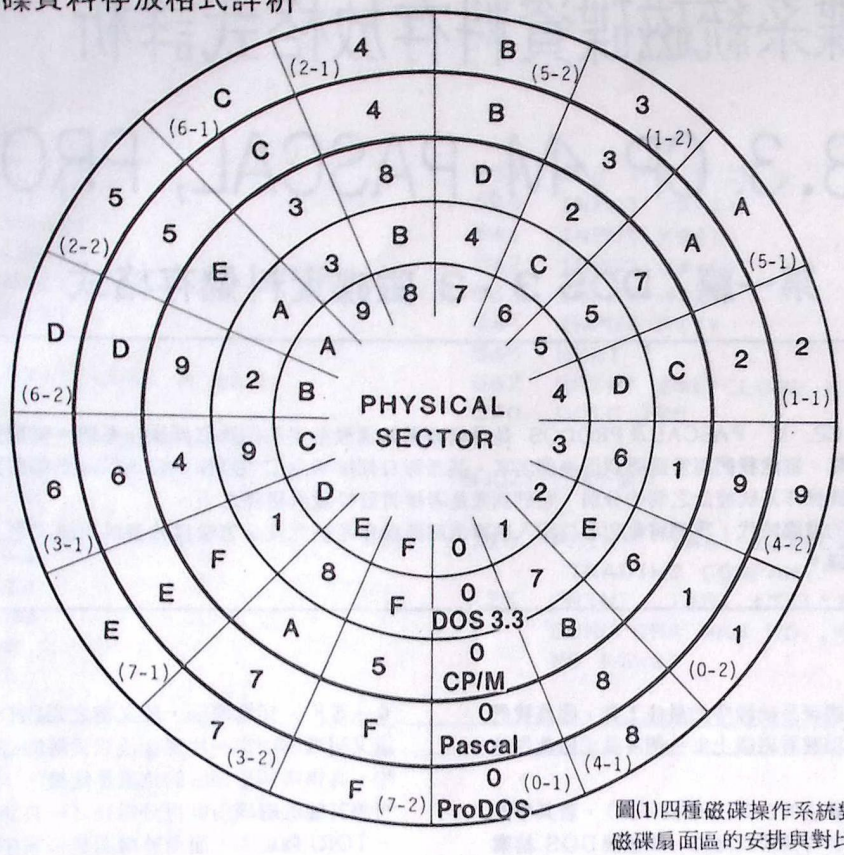
那麼你或許會追問，爲什麼那些天才的磁碟系統設計者不將磁碟扇區以順序安排？是否有「亞茂造餅」的意思呢！

答案當然不是，因爲ＤＯＳ 的設計者已顧及周邊硬件對資料處理的時間延誤之問題，所以磁碟表面的扇區編碼是絕不可能以順序安排的。以上皆是一些極爲深奧的技術上問題，本文亦不打算在此細加分析之。ＤＯＳ 的設計者却巧妙地將磁碟表面的扇區，根據硬件的時間延遲而編成圖１的邏輯扇區編碼系統。在不同的磁碟系統下，其磁碟扇面的編排是可提供最省時的資料處理程序。

有些專爲APPLE 磁碟系統而設的ＺＡＰ程式，它是已內置了一個邏輯與實物扇面（LOGICAL—PHYSICAL SE—CTOR）的翻譯副程式。所以你在DOS 3.3 下，是沒法直接讀取CP／M 磁碟系統下的資料檔案，這皆因兩種系統的LOGICAL—PHYSICAL 扇面不同之故。

參看圖１中的DOS 3.3 扇區之分佈，和以APPLE 的機械碼監督程式去觀看$BFB8起始記憶的資料，這便可得知DOS 3.3 處理邏輯扇面翻譯成眞正實物扇面的編碼與列表所在處。

```
*BFB8.BFC7
0 D B 9 7 5 3 1 E C A 8 6 4 2 F
DOS 3.3 sector
0 1 2 3 4 5 6 7 8 9 A B C D E F
```



圖(1)四種磁碟操作系統對磁碟扇面區的安排與對比

每當DOS 3.3 接收了一個要求讀取「第 7 個扇區」內資料的指令程序，它便會從地址＄BFB8 至＄BFC7 的扇面翻譯表中，找尋至第 8 個參值一「1」。同樣，驅使DOS 3.3 讀取「第 1 個扇區」內的資料，它便會由列表中倒查出來（意旨由PHYSICAL 碼轉爲LOGICAL SECTOR NO），這一點是會很容易打擾各位的邏輯思維。

蓋因DOS 3.3 的設計者是將LOGICAL SECTOR （＄1）相比成PHYSICAL SECTOR 的＄D（第14個）扇區，相對CP／M 系統却將LOGICAL SECTOR ＄1 置在PH—YSICAL SECTOR ＄3 上，而APPLE PASCAL 和PR—ODOS 則將LOGICAL SECTOR ＄1 同置於PHYSICAL SECTOR ＄2 上。

再細心參看圖 1，PASCAL與PRODOS 的扇面編排是完全相同，所以兩種系統是可以不需扇區編碼翻譯的程序，而互相讀取資料檔案。

PRODOS 扇面分析圖中，你可發覺每個扇區都是由 2 個參數所代表：（0－1），（4－1）等，它們是PROD—OS的BLOCK （資料群）編碼，而PRODOS 是以 2 個扇區去儲存檔案和根據個別扇區的BLO CK NO 去連續倒數，由磁軌＄0 至磁軌＄0F 間的扇面＄0 到＄22，因爲每條磁軌是可分爲35個扇區。

索引和儲存格式

圖 1 中亦祇顯示了磁軌＄0 中的首 8 個BLOCK NO ，比方如PRODOS 扇區＄0 的BLOCK NO（0－1），它的意思是BLOCK 0 的首一半；扇區＄1 的（0－2）則指一BLOCK 0 的次一半。根據每個扇區的BLOCK NO，DOS 便可翻譯成某段磁軌扇面之編碼（倒查亦可以）。

圖2、3、4和5 是分別顯示出個別的磁碟系統是甚樣將整個檔案系統分析，解碼和處理的格式。相信此點爲各電腦同好渴望所知的。

在每個圖表中，其最頂部是由35條磁軌所分隔，而LO—GICAL 扇面編碼則顯示於列表的左手面。

圖 2 中更清楚列明 DOS 3.3 系統的檔案處理，而圖3、4 和 5 則着重於CP／M、PASCAL與PRODOS 的BLO—CK NO。

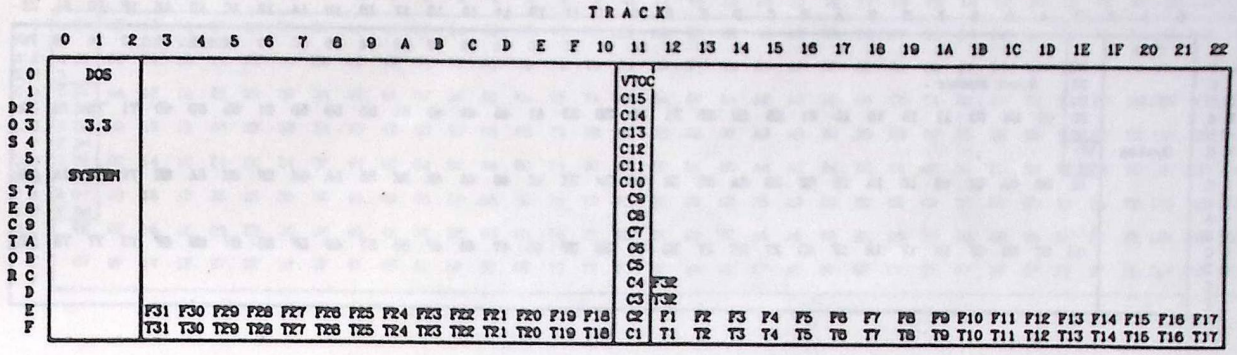
DOS 3.3與CP／M 操作系統的操作碼是由「SYSTEM」的字眼所代替，而PASCAL和PRODOS 則爲「LOAD」。

參看圖 2 中，你可發覺DOS 3.3 是佔用了＄0－＄2（共 3 條）的磁軌，其中的磁軌＄0 是完全空白的。對比CP／M 則全原儲錄於磁軌＄0 至＄2 上。還有，CP／M 是可以在沒有磁碟系統下去格式一片空白的資料磁碟。對比具有CP／M 系統與已格式的資料磁碟，後者可多出12K BY—TES 的資料儲存量。

附加一句，以上增加資料磁碟容量的技巧是祇容許在—MICROSOFT 最新的CP／M 2.23硬件系統上，因爲較早期的CP／M 2.20 SOFTCARD 是沒有此項功能的。

對比DOS 3.3與CP／M的系統，（SYSTEM）固定於＄0 至＄2 磁軌，而PASCAL與PRODOS 的系統（LOAD）則佔了磁軌＄0 上的頭 4 個扇面處。基本上SYSTEM 與LOAD 的功能一樣，它是在電腦啓動磁碟機時，將儲於磁碟上的磁碟操作系統讀入電腦記憶中，所以一張祇格式的資

圖(2) DOS 3.3 磁碟分佈圖



VTOC＝VOLUME TABLE OF CONTENTS
C1－C15：CATALOG SECTOR每個SECTOR 可容7個檔案名
15個SECTOR 共有最多105個檔案
T1－T32 ：每個檔案的TRACK／SECTOR 列表（每個T／SL SECTOR有122個檔案）

F1－F32 ：每個檔案的第一個SECTOR
（F1－F32顯示檔案存放次序，若磁碟有空間）
（超過1個SECTOR長度的檔案是會存放於同一磁軌上SECTOR E至0的位置上）
（檔案若超過1條TRACK長，則會依T1－T32 的次序找尋空間存放）

DOS 3.3 CATALOG SECTOR
首11個BYTES是每個CATALOG SECTOR的HEADER
110E是下一個CATALOG SECTOR，即TRACK 11，SECTOR 0E

檔案輸入：35BYTE／檔案

```
00110E0000000000000000
120F02C8C5CCCCCFA0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A00200

!- -!-!H E L L O - - - - - - - - - - - - - - - - - - - - - - - - - - !!- -
^   ^ ^---------------------- 檔案名字，最長不超過30字元 ----------------------^^
```

T／S 即
T＝12 （TRACK 12）
S＝0F （SECTOR 0F）
假如被DELETE，T會＝00

檔案類型：
00＝TEXT FILE
01＝INT BASIC
02＝APPLESOFT
04＝BINARY
08＝S類型FILE（DOS 3.3並未有用到）
10＝RELOCATABLE FILE
20＝A型FILE
40＝B型FILE
80＝FILE被LOCK上

檔案長檔，LO／HI BYTE
0200＝0002＝2 SECTOR

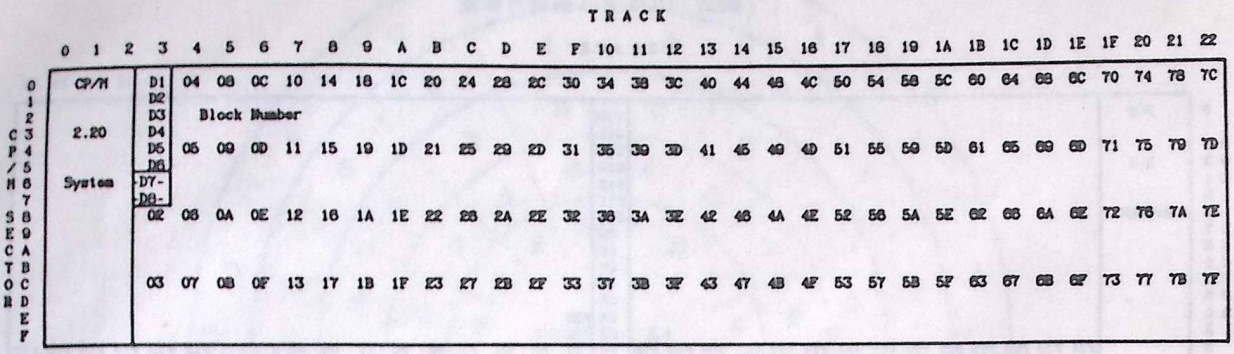
料磁碟是不會被BOOT動的。

細看圖2，3，4和5，你可發覺有些空置磁軌是用來儲存磁碟上檔案的名稱。在DOS 3.3下，你祇須鍵入CATALOG便可查看磁碟上的檔案名稱；反之在CP／M，PAS－CAL和PRODOS則爲DIRECTORY（PRODOS還可兩者並用呢！）。其實，電腦在接收了查閱檔案名稱時，它便依個別的磁碟系統格式下，去讀取磁碟上某段磁軌，而這些磁軌是已預置爲檔案名稱的儲存地方。

DOS 3·3

大部份的DOS 3.3 系統格式和副程式，你們可在DOS 3.3 用家手冊中找到，所以本文亦祇着重討論DOS 3.3 的

圖(3) CP／M 磁碟分佈圖



檔案處理和相容PRODOS檔案的問題。

在圖2中，磁軌＄11中的扇面＄0是被DOS 3.3 用作內容讀表的用途（VTOC—全寫爲VOLUME TABLE OF CONTENTS）。在磁軌＄11中的其餘扇面則完全用作檔案名稱的儲存，參閱圖中扇區＄0F的「C1」是代表第一個CATALOG扇面區，而其餘部份（C2—C15)則以倒序排列。

每一個CATALOG扇面起始的11BYTES 是代表了讀取檔案的前奏曲，緊接其後的第2和3個是儲存了下一個CA—TALOG 扇面的磁軌與扇面之所在。但你可發覺DOS 3.3的對下一個CATALOG扇面的參值與現時的扇面距離均是相同的空間長度，如磁軌＄11的第＄F CATALOG扇面之緊接扇區則爲＄0E和＄0D等。

圖2的低部，你亦可看到DOS 3.3 對每個檔案的參值是可爲35 BYTE的長度，其首2 BYTES（看圖2的＄12與＄0F)是指示某檔案是儲存於某個磁軌的扇面上。第3個BYTE是代表檔案的種類，而其餘的30BYTES 則爲檔案的名稱。相對最後的2個BYTES 是儲存檔案佔磁碟扇面的長度。平均來說，一個CATALOG扇面是可儲存達7個（35 BYTES)的CATALOG參值，亦即是11BYTE 的前奏加上7乘以35BYTES 長的參值，是剛等如256 BYTES（一個扇面的資料容量）。

談回檔案與CATALOG 的關係，圖2中的磁軌＄12 之扇面＄F的「T1」，是代表第一個儲存檔案的磁軌／扇面列表。比方以HELLO程式去將DOS 3.3 放於磁碟上的程序，它是被放於磁軌＄12的第一個扇區上，而其扇面＄E 則以「F1」代表（參看圖2）。代表以上的HELLO 程式的磁軌／扇面列表內容是由DOS負責解碼並分析如下：

```
00 0000000000000000000000000
0C 120E000000000000000000000
18 000.....................
...........
```

以上參值中的＄12和＄0E是代表了HELLO 程式是儲存於磁軌＄12的＄0E扇區上。若HELLO 程式是長於一個扇面區，則DOS 是會在磁軌＄12的扇區＄0E 起倒序地寫入＄0D，＄0C，＄0B……至＄0的扇面上。

若寫入的程式是長於35個扇面，DOS 便將檔案的其餘部份收錄於磁軌＄13上，過程與剛才的無異。

又假若HELLO 程式已一早儲存於磁碟上，而現今却被讀出並作出修改（增加了長度），和重新以同樣的檔案名寫回磁碟上，DOS 是會以磁軌＄12爲起始儲存點，而程式多出的部份則會智慧地被安排於其他空白的磁軌之扇面上。附加一句，以上的情況是祇局限於DOS 3.3的SAVE與WRI—TE指令程序。

圖2中是展示出DOS 3.3 在磁碟上儲存了32個檔案，而每個檔案的長度爲1個扇面。檔案F1至F17是儲存於磁軌＄12至＄22上。檔案F18 則存於磁軌＄10，而其餘檔案是由磁軌＄10倒序地儲存，直至磁軌＄3止（亦即代表磁碟已完全載滿檔案了）。

至於一些隨機儲存檔案，DOS 3.3 是會在OPEN 與CLOSE 指令下製造不同的隨機檔案，而一些順序檔案是可以APPEND的DOS 指令去將整個順序文字檔案併合，好使DOS 3.3 減短了讀取和找尋數個文字檔案的時間。

以下便要向各位介紹一種較快的文字檔案取存的技巧，用者可預先在DOS 3.3 下建立一個完全沒有資料的文字檔案。這種造法是方便將DOS 3.3 的檔案轉載於PRODOS系統上，原因DOS 3.3 在格式一片磁碟時是載入＄00的數值於每條磁軌上，而DOS 3.3和PRODOS對＄00的解碼是代表檔案完結（END OF FILE）的意思。

這樣，PRODOS上的CONVERT （檔案轉載）程式便可容易地將DOS 3.3 上的檔案轉載至PRODOS 磁碟系統上。

爲了使各讀者更容易了解DOS 3.3 的系統格式，以下便是一個技巧的簡介：

(1) 閱讀檔案名稱扇面C1—C15 是在磁軌＄11上，由扇面＄F 開始倒序至＄0，直至找到所需檔名爲止。若你擁有的ZAP程式是與DOS 3.3 不相容的話，你便要參考圖1的分析。

(2) 當你以ZAP程式找到所需檔案名時，首3個BYTE是磁軌／扇面參值，而其餘則爲檔名。

(3) 利用圖1去翻譯磁軌／扇面的參值。

(4) 以ZAP 程式根據參值指向的磁軌與扇面編碼去讀取指定檔案。

下期我們會爲大家剖析CP／M的磁碟資料儲存方式。

圖(4) PASCAL 磁碟分佈圖

TRACK

PASCAL SECTOR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22
0	L O A D	00	08	10	18	20	28	30	38	40	48	50	58	60	68	70	78	80	88	90	98	A0	A8	B0	B8	C0	C8	D0	D8	E0	E8	F0	F8	100	108	110
1		01	09	11	19	21	29	31	39	41	49	51	59	61	69	71	79	81	89	91	99	A1	A9	B1	B9	C1	C9	D1	D9	E1	E9	F1	F9	101	109	111
2	D I R E C T O R Y	D1 D2 D3 D4 D5 D6 D7 D8	0A	12	1A	22	2A	32	3A	42	4A	52	5A	62	6A	72	7A	82	8A	92	9A	A2	AA	B2	BA	C2	CA	D2	DA	E2	EA	F2	FA	102	10A	112
3			0B	13	1B	23	2B	33	3B	43	4B	53	5B	63	6B	73	7B	83	8B	93	9B	A3	AB	B3	BB	C3	CB	D3	DB	E3	EB	F3	FB	103	10B	113
4			0C	14	1C	24	2C	34	3C	44	4C	54	5C	64	6C	74	7C	84	8C	94	9C	A4	AC	B4	BC	C4	CC	D4	DC	E4	EC	F4	FC	104	10C	114
5			0D	15	1D	25	2D	35	3D	45	4D	55	5D	65	6D	75	7D	85	8D	95	9D	A5	AD	B5	BD	C5	CD	D5	DD	E5	ED	F5	FD	105	10D	115
6		06	0E	16	1E	26	2E	36	3E	46	4E	56	5E	66	6E	76	7E	86	8E	96	9E	A6	AE	B6	BE	C6	CE	D6	DE	E6	EE	F6	FE	106	10E	116
7		07	0F	17	1F	27	2F	37	3F	47	4F	57	5F	67	6F	77	7F	87	8F	97	9F	A7	AF	B7	BF	C7	CF	D7	DF	E7	EF	F7	FF	107	10F	117

BLOCK NUMBERS

圖(5) PRODOS 磁碟分佈圖

TRACK

ProDOS SECTOR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22
0	L O A D	00	08	10	18	20	28	30	38	40	48	50	58	60	68	70	78	80	88	90	98	A0	A8	B0	B8	C0	C8	D0	D8	E0	E8	F0	F8	100	108	110
1		01	09	11	19	21	29	31	39	41	49	51	59	61	69	71	79	81	89	91	99	A1	A9	B1	B9	C1	C9	D1	D9	E1	E9	F1	F9	101	109	111
2	D I R E C T O R Y	D1 D2 D3 D4 D5 D6 D7 D8	0A	12	1A	22	2A	32	3A	42	4A	52	5A	62	6A	72	7A	82	8A	92	9A	A2	AA	B2	BA	C2	CA	D2	DA	E2	EA	F2	FA	102	10A	112
3			0B	13	1B	23	2B	33	3B	43	4B	53	5B	63	6B	73	7B	83	8B	93	9B	A3	AB	B3	BB	C3	CB	D3	DB	E3	EB	F3	FB	103	10B	113
4			0C	14	1C	24	2C	34	3C	44	4C	54	5C	64	6C	74	7C	84	8C	94	9C	A4	AC	B4	BC	C4	CC	D4	DC	E4	EC	F4	FC	104	10C	114
5			0D	15	1D	25	2D	35	3D	45	4D	55	5D	65	6D	75	7D	85	8D	95	9D	A5	AD	B5	BD	C5	CD	D5	DD	E5	ED	F5	FD	105	10D	115
6	bit map		0E	16	1E	26	2E	36	3E	46	4E	56	5E	66	6E	76	7E	86	8E	96	9E	A6	AE	B6	BE	C6	CE	D6	DE	E6	EE	F6	FE	106	10E	116
7		07	0F	17	1F	27	2F	37	3F	47	4F	57	5F	67	6F	77	7F	87	8F	97	9F	A7	AF	B7	BF	C7	CF	D7	DF	E7	EF	F7	FF	107	10F	117

BLOCK NUMBERS

訂閱

（請用英文填寫，方便電腦處理）　（只適用於本港訂戶）　信封面請註明「訂閱」

本人欲訂閱

□電腦時代一年（十二期）HK$120

□電腦時代＋CompuHouse磁碟一年（十二期）HK$475

由第___期開始

姓名：_____

地址：_____

電話：_____

附上劃線支票 No. _____

支票收款人請填寫 “Computing Age Publisher”

請寄九龍中央郵箱71193號

NO. _____

蘋果老鼠仔的操作秘密

揭開老鼠仔神秘的一面

美國蘋果電腦公司繼 LISA 及 MAC後，於去年初又推出蘋果二型「老鼠仔」及一種名爲MOUSE PAINT的軟件。此類「老鼠仔」，包括有「老鼠仔」繪圖軟件，可在二型電腦上指示游標位置，選擇各種功能，指示電腦進行有關工作，如設計圖表、繪圖及其他供編寫報告或展示陳述用的視像輔助，用家可將文字插入圖畫內，亦可選擇多種不同的字體及顏色。至於軟件方面也相繼地不斷推出。

顧名思義，本文之內容是發掘二型「老鼠仔」之一些秘密及使用技巧。而在進入正題之前，我先談談一些裝配上的問題。

與其他介面咭一樣，蘋果二型「老鼠仔」介面咭原則上是可以插在任何一個 SLOT 上。但是請注意，蘋果電腦公司建議我們把此介面插在 SLOT 4，因爲 SLOT 0 是保留予記憶咭或80字行咭的。而傳統上，SLOT 1 是供印字機使用的，SLOT 2 是供 MODEM 咭的。SLOT 3 通常是予80字行咭的。至於 SLOT 6 ，衆所周知，是予磁碟機的。而 SLOT 7 RGB 彩色監察器所用的是予介面及 SUPER SPRITE 咭用的。

當我們把「老鼠仔」接駁好，會改變記憶體中$Cn0C及$ CnFB兩個地址（$代表16進制下之地址，n代表SLOT NO.由1～7）。把 $Cn0C 之值轉爲$ 20，$CnFB 爲 $D6 。這裡提供一個簡單的 BASIC 程式供各位找出 APPLE MOUSE介面之 SLOT位置。但注意：此程式只不過是核心，是主要部份而已。（參考程式一）

「老鼠仔」在畫面上是用2－D顯示的（TWO－DIM－ENSIONAL CARTESIAN SYSTEM）。其起點是（0，0），其X，Y坐標之距離是1023。在此你會發現，畫面結構是與在搖控桿（JOYSTICKS）下不同，在搖控桿或 PADDLES 下，X，Y位置是0～255。

但是畫面長度爲1023時，會出現一種轉譯上的困難。在高解像下，蘋果電腦在解像度下是橫面280點，直則192點（在 APPLE Ⅱc 下則有雙重解像，故是例外）。而蘋果電腦是如何解決這問題呢？

```
10 SLOT = 0: GOSUB 1000
20  IF SLOT = 0 THEN  PRINT "CANN
    OT FIND MOUSE INTERFACE CARD
     !" END
.
.
.
1000  FOR S = 0 TO 6
1010  IF  PEEK ( - 16116 + (S * 2
    56)) = 32 AND  PEEK ( - 1587
    7 + (S * 256)) = 214 THEN SL
    OT = S + 1: RETURN
1020  NEXT S
1030  RETURN
```

程式 1

其實這問題只是涉及到簡單數學的：－

HORIZONTAL ＝289／1024＝0.273

VERTICAL＝192／1024＝0.188

「老鼠仔」事實上每移動一格（INCREMENT）則表示在橫面的動高解像下之 0.273 點或直面之 0.188 點。當然，由於電腦畫面並不理會不完整之點，故你可使用 INT指令去產生一個整數之值。

首先，假定你根據前面所說，把老鼠介面咭放在 SLO－T 4，則程式二會幫你測出老鼠仔現況之有關位置。

但注意：如果在你使用這程式時，是第一次使用老鼠仔的話，則你會得到錯誤訊息，這時按〔 RESET 〕是你唯一的選擇。

對於老鼠仔上之按扭應注視，因其是作爲一個方向指示標（DIRECTIONAL CLUE）的工作。在程式二中，變數B是用作存放不同按扭之不同值。

表A告訴你按扭之值（變數B之值）。當你使用老鼠仔時，你會 CLICK、PRESS 或 DOUBLE CLICK其按扭。<註一>

註一：CLICKING 指你開啓老鼠上頂之按扭，使之開或關，去選擇一功能或目錄顯示。

蘋果老鼠仔的操作秘密

表(A)

Value	Now	Last Time
1	Pressed	Pressed
2	Pressed	Released
3	Released	Pressed
4	Released	Released

例如，當老鼠仔連行至程式二中之 LINE 210 四次時，則B之值是3，2，4及4，此時，你可肯定按扭是CLICK 的。卽是說，你按下了按扭（不一定是按下，可能按上——回復正常），代表值之3，但當再按時，表值之2，而當保留現況時，則表4，接着之值是4。

如當你使用 DOUBLE CLICK 時，則值會是3，2，3，2。

當然，你不能單獨依賴老鼠仔工作。在使用老鼠仔時，你可能會失去施展你的打字技術之機會，也可能令你的手漸漸「廢了」。

如你要使用電腦寫一封信時，如利用老鼠仔幫助時，此種工作稱爲 SHIFT－CLICKING，但，這也需要鍵盤的幫助。總之，鍵盤與老鼠仔合併工作時是最好不過的。當你在鍵盤上按一鍵，則例子中變數B會變爲負數，而代表不同按迅動之數目則不變。他們是會被合併並乘以－1。

但變數B之值是不會變回正數的，除非你按＜RESET＞並 POKE－16368，0。

至於其他有關之代碼是非常容易明白的。

當你使用文字幕時，你必須緊記，此時你所用的字元全是用點表示的。

```
100  PRINT  CHR$ (4);"PR#4": PRINT
     CHR$(1)
110  PRINT  CHR$ (4);"PR#0"
120  REM  THESE TWO LINES INITIAL
     IZE THE MOUSE CARD, SET THE
     HORIZONTAL AND VERTICAL COOR
     DINATES TO 0,0, AND THEN RET
     UEN TO THE SCREEN. CHR$(1) I
     S THE INITIALIZATION CHARACT
     ER. WHEN YOU'RE DONE, FOLLOW
      THE SAME PROCEDURE, BUT SEN
     D CHR$(0) INSTEAD.
  :
  :
200  PRINT  CHR$ (4);"IN#4"
210  INPUT "";X,Y,B
220  REM  FIRST, YOU OPEN THE CAR
     D FOR READING WITH IN#4.THEN
     , YOU USE THE INPUT STATEMEN
     T TO GET THE HORIZONTAL (X)
     AND VERTICAL (Y) COORDINATES
     . B IS THE STATUS OF THE BUT
     TON. USING DOUBLE QUOTE MARK
     S PREVENTS THE APPLE FROM
           SENDING A QUESTION MA
     RKS (?) TO THE CARD AS IT US
     UALLY DOES WHEN YOU USE AN I
     NPUT STATEMENT. THE CARD ALW
     AYS RESPONDS WITH THESE THRE
     E VALUES WHEN YOU READ FROM
     IT.
```

程式2

好了！希此文能使你對你的老鼠仔有更深的認識。■

PDL的有趣用途

大家都知道，在BASIC中PDL這個指令是用來計算遊戲棒的阻值。它的重要性，可以在用BASIC寫的遊戲中看到。但是，你又知否它一些有趣的功能呢？

PDL（28—43）

當你觸動它時（例如X＝PDL（28）或PRINT PDL（28）），便會把整個系統鎖著，就算按RESET鍵也不能重獲控制權，除非關掉電源，重新開機。

PDL（204—219）

這會把揚聲器觸動，傳出「卡搭」的聲响。這有如觸動$C030一樣。例如打入

FOR I＝1 TO 1000：PRINT PDL（204）：NEXT

便可從揚聲器聽到聲音。

PDL（236—239）

觸動這些位置，就有如觸動－16540＋N 的位置。

POKE－16304，0＝PRINT PDL（236）＝設定圖形顯示

POKE－16303，0＝PRINT PDL（237）＝設定文字顯示

POKE－16302，0＝PRINT PDL（238）＝設定全頁顯示

POKE－16301，0＝PRINT PDL（239）＝圖形，文字混合顯示 ■

仿效MOSSE PAINT而設計的實用工具軟件

APPLE PAINT 繪圖工具

曾經接觸過MAC PAINT或MOUSE PAINT的朋友會認識到這兩個軟件對於繪圖的能力，其內置的各種幾何圖形、花邊、底色、文字再加上「用家友善」的設計，繪畫任何圖案設計實是易如反掌。可惜II PLUS或相容電腦的用家無緣使用。雖然蘋果公司推出了MOUSE，並有一個MOUSE PAINT 軟件，但接近一千元的一張週邊咭並非每個用家所願意負担。爲了讓大家能獲得與擁有MOUSE或使用MACINTOSH的朋友相同的樂趣，筆者嘗試編寫一個類似模擬程式，並利用歡樂棒作爲模擬老鼠仔。各位可任意設計出想像中的圖案。

本程式名字叫APPLE PAINT，除主程式APPLE PAINT外，尚有兩個圖形表，亦是本軟件的靈魂所在。一個圖 表名爲PAINT PATTERN ，收藏有多種的幾何圖形，另一個是CHARACTER ，主要是英文字母字形表，供高解像圖解說用。

大家可從表A中看到PAINT PATTERN 圖形表內的各種圖形。APPLE PAINT 主程式就是利用到這些圖形，閣下可將歡樂棒視作爲一支畫筆，只要選擇適當的圖形，便可隨意在螢幕上任何地方畫上該圖形。圖形的選擇是用A至Z鍵（見表B），然後用PADDLE BUTTON 0 畫出。本程式的繪畫速度相當快捷，每秒可繪出十個圖形。假如閣下要慢工出細貨，可利用SPACE BAR來做，只要在APPLE PAINT主程式中加上一句：

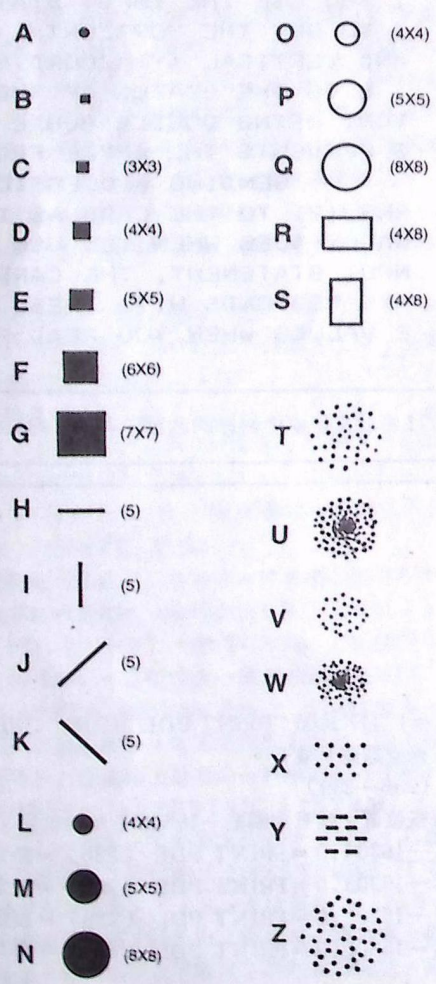
9003 IF PK＝160 THEN XDRAW D AT X%，Y%：RETURN

APPLE PAINT 程式除能夠繪出各種圖形外，還可畫出直線或間斷線、除去圖形、選擇DRAW或XDRAW模態，印點、選圖、以垂直或水平形式加上文字說明（文字有大小楷選擇）。表C是本程式的功能指令表。而表D所示是繪畫指令。

筆者編寫此程式時，其中一項原則是要「用家友善」，故此，閣下不必要擁有太多的電腦操作常識，也可輕易使用。

大家從表C裏各項功能指令中可看到，選擇(1)是繪圖。在此模態下，要畫一條線，按下〔•〕鍵（句號）設定線的開端，然後利用控制棒移動游標至適當的位置，然後按〔—〕鍵或〔;〕鍵來畫出一條直線或間斷線。假如要繼續畫線，只要將游標移至另一點即可。

表(A) 圖形表裏各種的圖形，大家可按下A至Z任何一個鍵即可。



表(B) APPLE PAINT的圖形選擇指令：

A	點	N	球體(8)
B	方塊（2×2）	O	圓形(4)
C	方塊（3×3）	P	圓形(5)
D	方塊（4×4）	Q	圓形(8)
E	方塊（5×5）	R	水平長方形
F	方塊（6×6）	S	垂直長方形
G	方塊（7×7）	T	見圖A
H	水平線	U	見圖A
I	垂直線	V	見圖A
J	東北方向對角線	W	見圖A
K	西北方向對角線	X	見圖A
L	球體(4)	Z	見圖A
M	球體(5)		

APPLE PAINT 繪圖工具

表(D) 繪劃指令

—	從最後一點游標位置開始劃線
;	從最後一點游標位置開始劃間斷線
•	開始劃直線或繪點
@	抹擦螢幕
RETURN	返回主索引（即表C）
ESC	觀看指令
BUTTON＃0	繪圖形
KNOB＃0	設定游標水平位置
KNOB＃1	設定游標垂直位置
1—7	選擇顏色

表(C) 功能指令表

0	終止
1	繪圖
2	消圖
3	印字機輸出圖畫
4	觀看磁碟CATALOG
5	加插文字標記
6	將圖畫SAVE入碟
7	從磁碟取出圖畫
8	抹去圖形
9	觀看螢幕
C	觀看各指令

表(E) 高解像頁色彩

第一組		第二組	
0	黑1	4	黑2
1	綠	5	橙
2	紫	6	藍
3	白1	7	白2

要消除螢幕某部份，按下〔@〕鍵，然後按照螢幕指示去做即可。此項功能主要是將要被清洗的部份抹上黑色。大家可從表E裏看到蘋果高解像頁色彩的決定，總共分兩組。大家留意，在配色時，千萬不要讓第一組的色彩與第二組的色彩作水平並排一起，這會破壞了畫面的和諧。大家只要按下數字0至7任何一個鍵就可選擇顏色。

DRAW或XDRAW（：鍵）能讓你連續繪畫選定的圖形，而連續繪畫的時間視乎BUTTON 0按下的時間。大家只要多加練習就能掌握。

當要在圖畫上加上文字說明，可按功能指令表（表C）中的5，首先要選擇垂直或水平書寫，然後便可開始。文字可以是大楷或小楷字，大家於寫字之前用ＥＳＣ鍵來選擇。在大楷模式時，說明是以反白模式顯示。

總括來說，這是一個使用家非常有用的程式，使用之時並沒有任何困難之處，不過筆者要強調一點，在繪圖時，最好能用一個有微調的控制棒，這樣效果會好很多。

本程式共有四個列表。列表1是主程式APPLE PAINT。列表2是26個圖形表PAINT PATTERN PE，各位打入後可用下列指令存入磁碟。

BSAVE PAINT PATTERN，A$800，L$200

列表3是大小楷字母圖形表CHARACTER，大家打入此表後，可用下列指令存入磁碟：

BSAVE CHARACTER，A$1000，L$500。

各位要打入列表2及列表3兩個機械碼列表，可先CA—LL—151〔RETURN〕進入監督程式，然後照每行的數值打入。例如第一行

1000— 7E 00 FE 00 00 01 06 01.

各位可依下列方法打入：

1000：7E 00 FE 00 00 01 06 01

假如各位還未懂得，可參照第三期〔如何輸入機械語言〕。

列表4是一個HELLO程式，名爲PAINT HELLO。其主要的作用是設定HIMEM位置，不讓程式劃花記憶。

各位在執行本文程式時，必須先RUN PAINT HELLO，切記。■

列表(1)

```
**********
** APPLE PAINT **
**      BY      **
** COMPUHOUSE   **
**********
0  POKE  - 16302,0
1  ONERR  GOTO 63990
2 HC = 3: ROT= 0: SCALE= 1:C = 3:
     HCOLOR= 3: POKE 230,32: CALL
     62450:S = 1
3 GZ = 1: GOSUB 9800
10 D$ =  CHR$ (4): PRINT D$"BLOAD
     PAINT PATTERN": POKE 232,0:
     POKE 233,8
15  PRINT D$"BLOAD CHARACTER,A256
     0"
20  GOTO 600
40  GOSUB 62000: RETURN
90  PRINT "": TEXT : HOME : PRINT
     "MOVE CURSOR TO UPPER LEFT C
     ORNER OF THE ERASE BLOCK & H
     IT SPACE. THEN MOVE THE CURS
     OR TO THE LOWER RIGHT CORNER
     OF THE ERASE BLOCK & HIT SP
     ACE.": GOSUB 63000: POKE  -
     16304,0: POKE  - 16297,0
91 X% = 1.094 *  PDL (0):Y% = .74
     9 *  PDL (1): XDRAW 2 AT X%,
     Y%: FOR QW = 1 TO 50: NEXT :
      XDRAW 2 AT X%,Y%
92 PK =  PEEK ( - 16384): IF PK >
     127 THEN  POKE  - 16368,0: PRINT
     "": GOTO 100
95 OO =  PEEK ( - 16336): GOTO 91

100 PX% = 1.094 *  PDL (0):PY% =
     .749 *  PDL (1): XDRAW 2 AT
     PX%,PY%: FOR QW = 1 TO 50: NEXT
     : XDRAW 2 AT PX%,PY%
102 PK =  PEEK ( - 16384): IF PK >
     127 THEN  POKE  - 16368,0: PRINT
     "": GOTO 110
105 OO =  PEEK ( - 16336): GOTO 1
     00
110  HCOLOR= 0:08 = PX%:01 = PY%:
     00 = X%:09 = Y%: FOR 00 = 00
      TO 08: HPLOT 00,09 TO 00,01
     : NEXT : HCOLOR= C: RETURN
```

APPLE PAINT 繪圖工具

```
220  IF  LEN (A$) = 0 AND A$ <  >
     " " THEN  RETURN
222  HCOLOR= 3
225 AA =  ASC (A$) - 31: IF AA =
     - 4 THEN UP =  NOT UP: GOTO
     370
228  IF UP = 0 AND AA > 33 AND AA
     < 60 THEN J = AA + 32: GOTO
     231
230 J = AA: GOTO 359
231  IF AA < 1 OR AA > 96 THEN 40

359  IF FX = 0 THEN  DRAW J AT X,
     Y,: HCOLOR= C: RETURN
360  XDRAW J AT X,Y: HCOLOR= C: RETURN

370  IF FV = 1 THEN Y = Y - 7: RETURN

380 X = X - 7: RETURN
600  TEXT : HOME : VTAB 2: INVERSE
     : HTAB 11: PRINT "MENU FOR A
     PPLE PAINT": NORMAL : PRINT
     : POKE 232,0: POKE 233,8
605  HCOLOR= C: ROT= 64: SCALE= S
     : CALL 54915:FF = 1:FX = 0:F
     V = 0
606  ONERR  GOTO 63990
609  PRINT "(0) QUIT"
610  PRINT "(1) PAINT PICTURE"
611  PRINT "(2) DELETEPICTURE"
612  PRINT "(3) PRINT OUT PICTURE
      ON PRINTER"
613  PRINT "(4) SEE DISK CATALOG"

614  PRINT "(5) INSERT EXTRA LABE
     LS"
615  PRINT "(6) SAVE PICTURE TO D
     ISK"
616  PRINT "(7) GET STORED PICTUR
     E FROM DISK"
617  PRINT "(8) ERASE & GIVE BACK
     GROUND COLOR": PRINT "(9) VI
     EW SCREEN"
619  PRINT : INVERSE : PRINT "C";
     : NORMAL : PRINT "= SEE COMM
     AN DS": PRINT
620  FLASH : PRINT "(0-9 OR C):";
     : NORMAL : GET AN$: IF  LEN
     (AN$) = 0 THEN 600
622  PRINT  CHR$ (13): CALL 1002
624  IF  ASC (AN$) = 67 THEN GZ =
     1: GOSUB 9800
625 Q =  VAL (AN$): IF AN$ = "0" THEN
      END
626  IF Q < 1 OR Q > 9 THEN 600
630  ON Q GOTO 1000,900,13000,170
     00,30000,31000,33000,37000,3
     2000
640  GOTO 600
900  PRINT : INVERSE : PRINT "HIT
      RETURN TO ABORT": NORMAL
902  PRINT : INPUT "FILE NAME TO
     DELETE: ";F$: IF  LEN (F$) =
     0 THEN 600
930 D$ =  CHR$ (4): PRINT D$"DELE
     TE";F$: GOTO 600
1000  POKE  - 16304,0: POKE  - 16
     297,0:D = 2
1010 X% = 1.094 *  PDL (0):Y% = .
     749 *  PDL (1)
1015  XDRAW D AT X%,Y%:PK =  PEEK
     ( - 16384): IF PK > 127 THEN
      POKE  - 16368,0: XDRAW D AT
     X%,Y%: GOSUB 9000: GOTO 1010

1025  IF  PEEK ( - 16287) > 127 THEN
      XDRAW D AT X%,Y%: GOSUB 104
     0: GOTO 1010
1030  XDRAW D AT X%,Y%: GOTO 1010

1040  IF DR = 0 THEN 1060
1050  XDRAW D AT X%,Y%:OX% = X%:O
     Y% = Y%: RETURN
1060  DRAW D AT X%,Y%:OX% = X%:OY
     % = Y%: RETURN
9000 C = HC: IF PK = 141 THEN  POP
     : GOTO 600
9001  IF PK = 192 THEN 90
9002  IF PK = 155 THEN 9100
9004  IF PK = 174 THEN D = 2: GOTO
     9200
9005  IF PK = 173 THEN D = 2: GOTO
     9300
9006  IF PK = 186 THEN 9030
9007  IF PK = 187 THEN D = 2: GOTO
     9700
9008  IF PK > 175 AND PK < 184 THEN
     9400
9009  IF PK < 193 THEN  RETURN
9010 D = PK - 192: IF PK > 192 THEN
      RETURN
9020  GOSUB 62000: RETURN
9030  HOME : POKE  - 16303,0: POKE
      - 16298,0:DR =  NOT DR:X$ =
     " DRAW": IF DR = 1 THEN X$ =
     "XDRAW"
9040  VTAB 9: PRINT "YOU'RE NOW I
     N ";: FLASH : PRINT X$: NORMAL
     : PRINT " MODE.": GOSUB 6300
     0: POKE  - 16304,0: POKE  -
     16297,0: RETURN
9100  TEXT : GOTO 9800
9200  HPLOT X%,Y%:OX% = X%:OY% =
     Y%: RETURN
9300  HCOLOR= C: HPLOT OX%,OY% TO
     X%,Y%
9310  IF  ABS (OX% - X%) <  ABS (
     OY% - Y%) THEN  IF C <  > 3 AND
     C <  > 7 THEN  HPLOT OX% + 1
     ,OY% TO X% + 1,Y%
9320 OY% = Y%:OX% = X%: HCOLOR= H
     C: RETURN
9400 HC = PK - 176:C = HC: HCOLOR=
     HC: RETURN
```

APPLE PAINT 繪圖工具

```
9700  IF  ABS (Y% - OY%) >  ABS (
    X% - OX%) THEN 9750
9705 OY = OY%:YI = (Y% - OY%) / (
    ( ABS (OX% - X%) / 3) + ( ABS
    (OX% - X%) = 0))
9710  HCOLOR= C: FOR OX = OX% TO
    X% STEP 3 * (X% > OX%) - 3 *
    (X% <  = OX%): HPLOT OX,OY:O
    Y = OY + YI: NEXT
9720 OY% = Y%:OX% = X%: RETURN
9750 OX = OX%:XI = (X% - OX%) / (
    ( ABS (OY% - Y%) / 3) + ( ABS
    (OY% - Y%) = 0))
9760  HCOLOR= C: FOR OY = OY% TO
    Y% STEP 3 * (Y% > OY%) - 3 *
    (Y% <  = OY%): HPLOT OX,OY:O
    X = OX + XI: NEXT : GOTO 972
    0
9800  HOME : INVERSE : HTAB 10: PRINT
    "COMMANDS": NORMAL : PRINT "
    - = LINE FROM LAST POINT PLO
    TTED": PRINT
9805  PRINT "; = DOTTED LINE FROM
     LAST POINT PLOTTED": PRINT
    : PRINT ".= BEGIN LINE HERE
    OR DRAW DOT": PRINT
9810  PRINT "@ = AREA ERASE": PRINT
    : PRINT ": = DRAW/XDRAW TOGG
    LE SWITCH": PRINT : PRINT "0
    -7 = CHOOSE COLOR": PRINT : PRINT
    "ESC = SEE COMM_ANDS": PRINT
    : PRINT "RETURN = BACK TO PR
    OGRAM MENU": PRINT
9820  PRINT "BUTTON #0 = PAINT WI
    TH SHAPE": PRINT : PRINT "KN
    OB #0 = HORIZONTAL COORDINAT
    E": PRINT : PRINT "KNOB #1 =
     VERTICAL COORDINATE"
9830  GOSUB 63000
9900  TEXT : HOME : PRINT "A = DO
    T": PRINT "B = 2X2 SQUARE C
    = 3X3 SQUARE": PRINT "D = 4X
    4 SQUARE E = 5X5 SQUARE": PRINT
    "F = 6X6 SQUARE G = 7X7 SQUA
    RE"
9910  PRINT "H = HORIZONTAL LINE"
    : PRINT "I = VERTICAL LINE":
     PRINT "J = N.E. DIAGONAL": PRINT
    "K = N.W. DIAGONAL": PRINT "
    L = BALL (4 WIDE)": PRINT "M
     = BALL (5 WIDE)": PRINT "N
    = BALL (8 WIDE)": PRINT "O =
     CIRCLE (4 WIDE)"
9920  PRINT "P = CIRCLE (5 WIDE)"
    : PRINT "Q = CIRCLE (8 WIDE)
    ": PRINT "R = HORIZONTAL REC
    TANGLE": PRINT "S = VERTICAL
     RECTANGLE": PRINT "T = LARG
    E PAINTBRUSH": PRINT "U = LA
    RGE TIPPED PAIN TBRUSH"
9930  PRINT "V = SMALL PAINTBRUSH
    ": PRINT "W = SMALL TIPPED P
    AIN TBRUSH": PRINT "X = DARK
    ER PATTREN": PRINT "Y = LIGH
    TER PAT TERN": PRINT "Z = VE
    RY LARGE SCATTERBRUSH"
9998  GOSUB 63010: IF GZ = 1 THEN
    GZ = 0: RETURN
9999  GOTO 1000
13000  HOME : VTAB 1: PRINT " NOW
     TYPE THE CTRL CHARACTERS YO
    U NEED TOGET YOUR PRINTER TO
     PRINT OUT HI-RES   PAGE 1 A
    S A GRAPHICS PICTURE. MAKE S
    URE YOU HIT RETURN AFTER YOU
    'RE DONE TYPING ALL THE CTRL
     CHARACTERS:": PRINT :A9 = 0

13005  FLASH : PRINT "DON'T FORGE
    T TO TURN ON YOUR PRINTER !"
    : NORMAL : PRINT
13010 A9 = A9 + 1
13020  GET CR$(A9): IF  ASC (CR$(
    A9)) <  > 13 THEN 13010
13021  PRINT  CHR$ (13): CALL 100
    2
13022  PRINT  CHR$ (4);"PR#1"
13025  FOR QW = 1 TO A9: PRINT CR
    $(QW);: NEXT
13028  PRINT  CHR$ (4);"PR#0"
13030  GOTO 600
17000  PRINT : PRINT  CHR$ (4);"C
    ATALOG": GOSUB 63000: HOME :
     GOTO 600
30000  GOSUB 36000:FZ = 0
30001 AN$ = "":Z$ = "": PRINT "LA
    BEL: ";
30002  GET Z$:AN$ = AN$ + Z$: IF
     ASC (Z$) = 13 THEN AN$ =  LEFT$
    (AN$, LEN (AN$) - 1): PRINT
     CHR$ (13): CALL 1002: GOTO
    30007
30003  IF  ASC (Z$) = 8 THEN AN$ =
     LEFT$ (AN$, LEN (AN$) - 1):
     IF  LEN (AN$) > 0 THEN AN$ =
     LEFT$ (AN$, LEN (AN$) - 1)
30004  IF  ASC (Z$) = 27 THEN FZ =
     NOT FZ: GOTO 30002
30005  IF FZ = 1 THEN  INVERSE
30006  PRINT Z$;: NORMAL : GOTO 3
    0002
30007  POKE 232,0: POKE 233,10: NORMAL

30010  HOME : VTAB 2: PRINT "USE
    PADDLES TO PADDLES TO POSITI
    ON LABEL; PRESS ANYKEY TO PR
    INT LABEL. USE ESC FOR UPPER
     AND LOWER CASE SWITCH.": GOSUB
    63000
30020  POKE  - 16304,0: POKE  - 1
    6297,0: HCOLOR= 3:UP = 0
30021  IF FV = 1 THEN  ROT= 16
30022 FX = 1: GOTO 30030
```

```
30024  IF FV = 1 THEN X = PO:Y =
     P1: FOR Q = 1 TO  LEN (AN$):
     A$ =  MID$ (AN$,Q,1): GOSUB
     35000: GOSUB 220:Y = Y + 7: NEXT
     :X = PO:Y = P1:UP = 0: FOR Q
      = 1 TO  LEN (AN$):A$ =  MID$
     (AN$,Q,1): GOSUB 35000: GOSUB
     220:Y = Y + 7: NEXT : GOTO 3
     0030
30025 X = PO:Y = P1: FOR Q = 1 TO
      LEN (AN$):A$ =  MID$ (AN$,Q
     ,1): GOSUB 35000: GOSUB 220:
     X = X + 7: NEXT :X = PO:Y =
     P1:UP = 0: FOR Q = 1 TO  LEN
     (AN$):A$ =  MID$ (AN$,Q,1): GOSUB
     35000: GOSUB 220:X = X + 7: NEXT

30030 PO = 1.094 *  PDL (0):P1 =
     .749 *  PDL (1)
30031 UP = 0
30035 PP =  PEEK ( - 16384): IF P
     P > 127 THEN  POKE  - 16368,
     0: GOTO 30128
30040  GOTO 30024
30128  IF FV = 0 THEN X = PO:Y =
     P1: FOR Q = 1 TO  LEN (AN$):
     A$ =  MID$ (AN$,Q,1): GOSUB
     35000: GOSUB 220:X = X + 7: NEXT
     : GOSUB 63010: GOTO 600
30129 X = PO:Y = P1: FOR Q = 1 TO
      LEN (AN$):A$ =  MID$ (AN$,Q
     ,1): GOSUB 35000: GOSUB 220:
     Y = Y + 7: NEXT : GOSUB 6301
     0: GOTO 600
31000  HOME : VTAB 1: INVERSE : IF
     F$ = "" THEN F$ = "NONE"
31002  PRINT "CURRENT FILE NAME:
     "F$: NORMAL : PRINT
31005  PRINT : INVERSE : PRINT "H
     IT RETURN TO ABORT. DO NOT T
     RY TO SAVE PICTURES ON THE P
     ROGRAM DISK.": NORMAL : PRINT

31010  INPUT "PICTURE NAME: ";F$:
      IF  LEN (F$) = 0 THEN 600
31040  PRINT  CHR$ (4);"BSAVE";F$
     ",A8192,L8192": GOTO 600
32000  POKE  - 16304,0: POKE  - 1
     6297,0: GOSUB 63010: GOTO 60
     0
33000  PRINT : INVERSE : PRINT "H
     IT RETURN TO ABORT.": NORMAL
     : PRINT
33010  INPUT "PICTURE NAME: ";F$:
      IF  LEN (F$) = 0 THEN 600
33040  PRINT  CHR$ (4);"BLOAD";F$
     ;",A8192": POKE  - 16304,0: POKE
      - 16297,0: GOSUB 63010: GOTO
     600
35000  IF X > 273 THEN X = 273
35005  IF Y > 184 THEN Y = 184
35010  RETURN
36000  HOME : VTAB 1: INVERSE : PRINT
     "HIT RETURN TO ABORT"
36001  NORMAL
36002  PRINT "HORIZONTAL OR VERTI
     CAL LABEL? (H/V):";: GET AZ$
     : PRINT AZ$: PRINT  CHR$ (13
     ): CALL 1002
36004  IF  ASC (AZ$) = 13 THEN  POP
     : GOTO 600
36005  IF  ASC (AZ$) <  > 86 AND
      ASC (AZ$) <  > 72 THEN 3600
     0
36010  IF  ASC (AZ$) = 86 THEN FV
      = 1
36020  RETURN
37000  HOME : VTAB 8: FLASH : PRINT
     "DANGER: THIS WILL ERASE ENT
     IRE SCREEN1": PRINT : PRINT
     : INVERSE : PRINT "HIT RETUR
     N TO ABORT:": PRINT "SURE YO
     U WANT TO ERASE? (Y/N): ";: NORMAL : GET AN$
37005  PRINT AN$: PRINT  CHR$ (13
     )
37010  IF  ASC (AN$) <  > 89 THEN
     600
37020  PRINT : PRINT "GIVE COLOR
     (0-7): ";: GET Q$:AQ =  VAL
     (Q$): IF AQ > 7 OR AQ < 0 THEN
     37020
37030  PRINT Q$: PRINT  CHR$ (13)
     : HGR : POKE  - 16302,0: HCOLOR=
     AQ: HPLOT 0,0: CALL 62454: HCOLOR= HC: GOTO 600
62000  FOR WQ = 1 TO 15:PK =  PEEK
     ( - 16336): NEXT : RETURN
63000  PRINT : PRINT "(PRESS SPAC
     E TO CONTINUE):";
63010 PH =  PEEK ( - 16384): IF P
     H > 127 THEN  POKE  - 16368,
     0: PRINT  CHR$ (13): RETURN

63020  GOTO 63010
63990  POKE 216,0:PQ =  PEEK (222
     ): PRINT "": PRINT "": PRINT
     ""
63991  ONERR  GOTO 63990
63992  IF PQ = 254 THEN  RESUME
63994  IF PQ = 6 OR PQ = 5 THEN  PRINT
     : PRINT "FILE NOT FOUND! TRY
      AGAIN...": GOSUB 63000
63999 C = 3: SCALE= 1: ROT= 64: GOTO
     600 ■
```

列表(4)

```
1  REM  ************************
2  REM  *                      *
3  REM  *   APPLE PAINT DRIVER  *
4  REM  *                      *
5  REM  *   COPYRIGHT 1985 (C)  *
6  REM  *   BY COMPUTING AGE    *
7  REM  *                      *
8  REM  ************************

10  POKE 104,64: POKE 16384,0
20 D$ =  CHR$ (4): PRINT D$"RUN A
     PPLE PAINT"
```

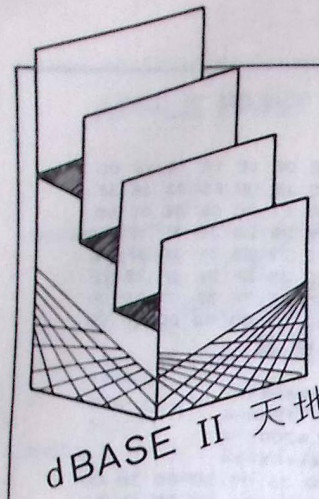
APPLE PAINT 繪圖工具

```
******************
*   CHARACTER    *
*  A$1000,L$500 *
******************
1000- 7E 00 FE 00 00 01 06 01
1008- 0C 01 1D 01 2B 01 37 01
1010- 44 01 48 01 50 01 58 01
1018- 66 01 6E 01 75 01 7A 01
1020- 7F 01 87 01 96 01 9E 01
1028- A9 01 B4 01 C1 01 CE 01
1030- DB 01 E4 01 F2 01 00 02
1038- 05 02 0C 02 15 02 1D 02
1040- 26 02 2F 02 3D 02 4C 02
1048- 5A 02 66 02 72 02 7F 02
1050- 8A 02 96 02 A2 02 AB 02
1058- B4 02 C1 02 C8 02 D5 02
1060- E3 02 EF 02 FB 02 08 03
1068- 17 03 24 03 2C 03 36 03
1070- 40 03 4C 03 58 03 64 03
1078- 6F 03 7C 03 83 03 90 03
1080- 98 03 9F 03 A4 03 AF 03
1088- BA 03 C4 03 D1 03 DC 03
1090- E5 03 F2 03 FD 03 05 04
1098- 0D 04 17 04 1F 04 2A 04
10A0- 36 04 40 04 4C 04 59 04
10A8- 64 04 70 04 77 04 7F 04
10B0- 88 04 92 04 9D 04 A9 04
10B8- B4 04 BF 04 C8 04 D2 04
10C0- D9 04 2D 25 2D 25 2D 2D
10C8- 14 65 2B 27 FD 0E 22 E7
10D0- 24 1E ED 1E 2A 75 12 1C
10D8- 6D 0E 6B 0E 6B E6 12 E7
10E0- 24 3F 22 66 6B 65 45 1C
10E8- 24 1E 2A 0C 12 2E 12 0E
10F0- 14 E7 24 1E 24 1E 24 1E
10F8- 6A 1C 07 75 0A 00 01 00
1100- 49 36 36 16 06 00 09 36
1108- 0D 24 04 00 09 36 36 36
1110- 0D 24 24 24 8D 13 16 1F
1118- 04 F8 16 06 00 49 2E B5
1120- F2 37 1C 67 61 3F 1C 0C
1128- 0E 05 00 31 25 4D F2 1E
1130- 1E 1E 4E 29 3C 07 00 09
1138- 0E F6 0E 0D 1E 0E 1F E7
1140- 24 20 04 00 49 36 06 00
1148- 49 1E 1E 36 0E 0E 05 00
1150- 49 0E 0E 36 1E 1E 1E 00
1158- 49 36 36 36 0D E0 04 60
1160- DF 73 16 1E 1E 00 49 32
1168- 36 2E 40 FF 3F 00 49 92
1170- 12 F6 06 00 2D 91 2A 2D
1178- 2D 00 49 92 92 06 00 91
1180- 92 0C 0C 0C 0C 05 00 09
1188- 2D 0E 36 36 1E 3F 1C 24
1190- 24 4E F1 1E 06 00 49 1E
1198- 35 36 76 3F 07 00 11 0C
11A0- 2D 0E F6 F7 1E 2E 2D 2D
11A8- 00 29 2D 35 1E 1E 75 F6
11B0- 3F 1C 04 00 49 31 36 0E
11B8- 37 3E 80 80 3F 64 0C 04
11C0- 00 29 2D B5 3B 3F B4 49
11C8- 31 F6 3F 1C 04 00 49 09
11D0- 3F 1E 1E 36 76 2D 0C E4
11D8- 3F 07 00 29 2D 35 1E 1E
11E0- 1E 36 06 00 09 2D 0E F6
11E8- 3F 1E 76 2D 0C 24 D8 1B
11F0- 24 00 09 2D 0E 36 F6 1E
11F8- 3F 04 40 2D DC 23 04 00
1200- 49 12 16 06 00 49 92 16
1208- F6 06 00 49 49 F1 1E 1E
1210- 0E 0E 0E 05 00 91 2D 2D
1218- 16 3F 3F 07 00 09 0E 0E
1220- 0E 1E 1E 1E 07 00 11 0C
1228- 2D 0E 1E 1E B6 06 00 09
1230- 2D 0E 36 96 3F E7 24 24
1238- 8D 36 25 04 00 91 36 36
1240- 4D 21 3C 3F 04 80 80 0C
1248- 0E 0E 36 00 29 2D 0E F6
1250- 3F 04 18 36 36 2E 2D 0C
1258- 24 00 49 89 1C 3F 1E 36
1260- 36 0E 2D 0C 04 00 29 2D
1268- 0E 36 36 1E 3F 27 24 24
1270- 04 00 29 2D B5 1A 3F 04
1278- 18 36 36 2E 2D 2D 00 29
1280- 2D B5 1A 3F 04 18 36 36
1288- 36 00 09 2D F5 DB 36 36
1290- 0E 2D 25 3C 07 00 31 36
1298- 36 6E 09 24 3C 3F 4C 21
12A0- 24 00 09 2D 1E 36 36 0E
12A8- 3F 07 00 49 09 36 36 F6
12B0- 3F 1C 04 00 31 36 36 6E
12B8- 09 1C 1C 1C 0C 0C 0C 0C
12C0- 00 31 36 36 2E 2D 2D 00
12C8- 31 36 36 6E 09 24 24 24
12D0- 1E 1E 1C 04 00 31 36 36
12D8- 6E 09 24 24 24 9F 13 0E
12E0- 0E 06 00 09 2D 0E 36 36
12E8- 1E 3F 1C 24 24 04 00 31
12F0- 36 36 2E 80 80 28 65 E4
12F8- 3F 04 00 09 2D 0E 36 FE
1300- 0E 0E 1F E7 24 24 04 00
1308- 29 2D 0E F6 3F 04 18 36
1310- 36 6E 09 1C 1C 05 00 09
1318- 2D 0E DF 33 0E 2D 0E F6
1320- 3F 1C 04 00 29 2D F5 33
1328- 36 36 06 00 31 36 36 0E
1330- 2D 0C 24 24 24 00 31 36
1338- 76 0E 0C 0C 24 24 04 00
1340- 31 36 36 66 0C AC 72 24
1348- 24 24 04 00 31 0E 16 1E
1350- 6E 09 E4 1C 0C 0C 24 00
1358- 31 0E 0E 36 2E 80 80 80
1360- 80 0C 24 00 29 2D 35 1E
1368- 1E 1E 1E 2E 2D 2D 00 31
1370- 36 36 2E 2D E5 1B 24 24
1378- 2C 2D 05 00 11 0E 0E 0E
1380- 0E 05 00 29 2D 35 36 36
1388- 3E 3F 67 09 24 24 07 00
1390- 91 12 0C 0C 0E 0E 05 00
1398- 91 92 12 2D 2D 05 00 09
13A0- 0E 0E 06 00 91 29 35 36
13A8- 2E DE 38 1C 0C 2D 00 31
13B0- 36 36 2E 70 65 24 1C F7
13B8- 06 00 91 49 39 3F 1E 36
13C0- 0E 2D 2D 00 49 09 36 36
13C8- 36 1C 1E E7 24 0C 75 06
13D0- 00 91 29 75 3E 3F 1C 36
13D8- 0E 2D 05 00 49 75 DF F6
13E0- 2D 1E 36 06 00 91 29 0D
13E8- 1E F5 35 F6 3F 04 08 E7
13F0- 24 00 31 36 36 2E 80 80
13F8- 60 75 36 36 00 49 16 77
1400- 36 1E 2D 05 00 49 16 77
1408- 36 F6 1C 04 00 31 36 36
1410- 6E E1 1C 0C 0C 04 00 09
1418- 35 36 36 1E 2D 05 00 91
1420- 36 36 0D 24 E4 0D 0E 36
1428- 36 00 91 36 36 0D 18 80
1430- 80 0C 75 36 36 00 91 29
1438- 75 36 1E 3F 1C 24 04 00
1440- 91 36 36 2E 80 80 2D 0C
1448- E4 3F 07 00 91 29 35 36
1450- 36 F5 03 80 80 38 1C 24
1458- 00 91 36 36 0D 18 80 80
1460- 0C 75 06 00 49 89 3A 3F
1468- 1E 0E 2D 0E 1E 3F 3F 00
1470- 49 F6 2D 1E 36 36 00 91
1478- 36 76 2D 0C 24 24 00 91
1480- 36 0E 0E 0C 0C 24 04 00
1488- 91 36 76 0C 24 95 62 24
1490- 24 00 91 0E 16 1E 4D E1
1498- 1C 0C 0C 0C 00 91 36 0E
14A0- 2D 25 24 95 D2 1E 3F 07
14A8- 00 91 2D 2D 1E 1E 1E 1E
14B0- 2D 2D 05 00 49 2D DE 73
14B8- 1E 77 F1 0E 2D 05 00 09
14C0- 0E 0E 0E 1E 1E 1E 1E 00
14C8- 29 75 1E 0E F5 73 1E 3F
14D0- 07 00 11 0C 0E 0E 0C 05
14D8- 00 29 2D 2D 35 3F 3F 3F
14E0- 2E 2D 2D 35 3F 3F 3F 2E
14E8- 2D 2D 35 3F 3F 3F 2E 2D
14F0- 2D 35 3F 3F 3F 07 00 00
14F8- 00 00 00 00 00 00 00 00
*       AA BD
```

列表(2)

```
******************
* PAINT PATTERN *
* A$800, L$200  *
******************
0800- 1A 00 36 00 38 00 3B 00
0808- 41 00 4A 00 58 00 6B 00
0810- 85 00 89 00 8D 00 93 00
0818- 99 00 A0 00 AD 00 C9 00
0820- CF 00 D9 00 E7 00 F3 00
0828- FF 00 27 01 64 01 72 01
0830- 84 01 99 01 B0 01 04 00
0838- 25 3F 00 3C 36 2D 24 04
0840- 00 65 3F 37 36 25 35 25
0848- 24 00 25 3F 36 2D 25 24
0850- 3F 3F 36 36 2D 2D 05 00
0858- 25 3F 36 2D 25 24 3F 3F
0860- 36 36 2D 2D 25 24 24 3F
0868- 3F 3F 00 25 3F 36 2D 25
0870- 24 3F 3F 36 36 2D 2D 25
0878- 24 24 3F 3F 3F 36 36 36
0880- 2D 2D 2D 05 00 2D 2D 05
0888- 00 24 24 04 00 0C 0C 0C
0890- 0C 0C 00 1C 1C 1C 1C 1C
0898- 00 25 27 37 37 35 2D 00
08A0- 25 3F 36 2E 2C 2E 20 E4
08A8- 3F 17 36 06 00 2D 25 27
08B0- 27 3F 37 37 37 36 35 35
08B8- 2D 25 25 25 3F 3E 3E 3C
08C0- 3C 2C 2C 2C 0E F7 2E 05
08C8- 00 21 1C F7 76 2D 00 09
08D0- E4 3F 1E 36 0E 2D 0C 04
08D8- 00 49 E4 1C 3F F7 1E 36
08E0- 76 0E 2D 65 0C 24 00 38
08E8- 3F 36 2E 2D 2D 2D 24 3C
08F0- 3F 07 00 21 24 3F 37 36
08F8- 36 36 2D 25 24 04 00 04
0900- 08 DC DF 1A 16 4E FA 1B
0908- 04 C0 04 40 4D B1 0A 4D
0910- 49 D6 D6 DE FB 1B DF D8
0918- 04 18 20 80 80 0C 80 80
0920- 4D 09 4D 51 4D 2A 00 35
0928- E7 2C 55 D6 DF E0 60 69
0930- 8D 16 D6 1F 18 1F 04 18
0938- 04 60 69 69 95 15 16 D7
0940- 1E 1F F0 07 F0 1F E0 38
0948- 80 80 60 60 09 4D 69 09
0950- 8E 31 8E B1 F3 3A DE FA
0958- FB 1B 4C DF 1B 1C 18 04
0960- 80 80 04 00 95 DF DC 04
0968- 08 4C 4D 8E 95 BA 1B DF
0970- 3B 00 35 3F 24 2D 0E 95
0978- DF 1F 20 60 4C 8D DF DB
0980- B3 56 06 00 4D 69 09 8D
0988- DF FB 1B 9F 4D 69 09 55
0990- DF FB 1B 9F 4D 69 09 55
0998- 00 2D 0D 2D 0D 2D 5E 1F
09A0- 3F 1F 3F 1F 2E 6D 2D 0D
09A8- 2D FE 3F 1F 3F 1F 07 00
09B0- 29 48 A9 12 97 1B DF FB
09B8- 18 18 04 18 18 18 DF F3
09C0- D3 1A 4E 09 8E FA DB 8E
09C8- 51 11 4D 01 08 08 96 0A
09D0- 4D 08 08 8D 51 05 08 80
09D8- 80 68 8A 61 80 80 20 80
09E0- 80 C0 04 80 80 18 DF 1B
09E8- 18 DF 9A 23 C0 1B DF DB
09F0- DF D3 9A 31 00 E0 64 00
09F8- 50 58 25 D0 31 2E 30 39
```

列表(3)



dBASE 入門篇

—— 初步認識

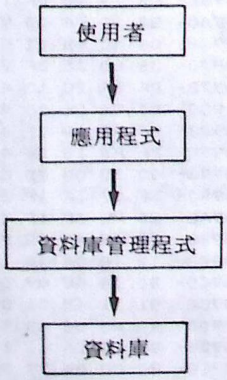
dBASE其實是DATA BASE的縮寫，而dBASE Ⅱ是一個資料庫管理系統或稱之爲DBMS（DATA BASE MANAGEMENT SYSTEM），它基本上是多組程式集合起來的程序，內置了命令組譯器（INTERPRETER），因此它可以算是一個程序，也是一種語言。

對大部份的資料庫管理程式而言，應用範圍都包括了總分類帳，存貨控制，應收帳款，查詢資料及電話名冊，郵寄列表系統，目錄製作，訂貨記錄，文獻目錄等。就本質上來說，舉凡任何要進行資料檔案建立，更新，分析和記錄的應用都包括在內，這就是DBMS的用途，不過絕大部份這類程式都是以BASIC，PASCAL，FORTRAN 或其它高階語言寫成，再呼叫資料庫管理程式和資料，然而，dBASE Ⅱ却用一種本身的內部語言，於是就不再需要任何其它語言了。

要理解dBASE 的最佳模式（MODEL）是將它想像成一個專門用來處理數據的文字處理器，文字處理器可以讓你建立任何文件及隨意修改編冊，而dBASE 則容許使用者隨意建立數據資料檔案及在有需要時可隨意更改資料。

DBASE 的結構

DBASE是資料庫管理系統DATA BASE MANAGEMENT SYSTEM的縮寫，它基本上是一組程式或程式集合體，將使用者連接到一個或更多資料集合體（即我們所稱的檔案）上，因爲DBASE是利用這個原理設計的，我們亦叫它作爲資料庫，它的結構流程圖如下：



相信很多朋友都對DBASE Ⅱ 這個名字都不會感到陌生，而且很多用戶都已對DBASE Ⅱ有著很深入的了解，也許閣下現時已對DBASE Ⅱ駕輕就熟，但無論如何，筆者現在由淺入深的次序去 探討它 如何爲你效勞。另一方面，對DBASE Ⅱ已經有著相當認識的朋友也可以利用本文作爲技巧上的檢討及作一個DBASE Ⅱ的綜觀，因爲在過去的幾年之中，各種新型的庫存管理系統都不斷地推出及快速成長，不過這類型的系統都相對地給DBASE Ⅱ比了下來，其原因是包括：

①DBASE Ⅱ是被當作一個關連式（RELATIONAL）的資料庫管理系統在市場上推出的。所謂關連式資料庫系統就是以二維空間表格（2—DIMENSIONAL TABLES）儲存資料（資訊）的系統，在二維表格中，資料可依儲存在表格中的相對關係存取。

②DBASE Ⅱ是一個以高級程式設計語言完成的檔案管理系統。這套系統語言十分易學，而且是結構化的，當各位熟悉之後，（甚至是程式設計師）都可以因此而採用DB—ASE 語言來快捷地發展程式，若果採用其它高級語言如FORTRAN或BASIC來編寫等效的資料庫管理程式時，效率會低得多及困難得多，例如我們在第十期刊登〔可任意訂定格式之資料記錄系統〕一文中的程式就是一個實例。

③DBASE可以維持程式及資料的獨立性。資料檔的結構可以在不用更動程式之下改變，程式也可在不改變資料檔案結構之下更動。

④資訊（及資料）是以單獨集合體或單獨檔案儲放，採用—RAM DOM ACCESS的檔案形式放置資料。不過CP／M作業系統的資料存取時間十分快，比起以DOS 3.3 形式存取的程式要快上十多倍。比著名的PFS快上8倍而且儲存量高十倍。

⑤資料可以被許多的應用程式使用（例如WORDSTAR，SUPERCALC 等），而在資料上可以經由其它公用程式來防止未經授權的存取資料。例如說，員工的薪水狀況不應該讓只需要員工地址的職員弄到手。

⑥DBASE 提供了在資料庫中排序，索引及列印資料的內置功能，使程式設計的發展工夫減到最低。

在未正式開始探討DBASE Ⅱ之前，要讓筆者介紹一些有關DBASE Ⅱ的歷史，本刊第十二期的dTEST 一文中提起過部份的歷史。DBASE Ⅱ的歷史跟它的能力同樣吸引人，它最初期的模式始於70年代，由位於加州（PASATINA）的專門研究太空科技的噴射推進實驗室（JET PROPULSI

什麼是DBASE II

—ON LABRATORY 簡稱JPL） 所發展，直到1974年，JPL 的科學家及工程師開始用一種高級資料庫管理系統來分析由許多JPL 無人太空船探測傳來的資訊。這個稱爲JPLD—IS（全名是JPL DATA MANAGEMENT AND INFOR—MATION RETRIEVAL SYSTEM）的資料管理及資訊擷取系統是JEP LONG 設計的，在大型電腦上運作。

WAYNE RATLIFF是實驗室中一位系統設計師，他想要一個能在他家中微電腦上運作的同類系統，他亦是JEP LONG的同事。RATLIFF 所著述的不僅僅是對這些十分有智慧的無人太空探測的控制，他還需要包含人工智能這門學問的整個領域，於是他開始利用JPLDIS 當作模式來撰寫某種人工智能的軟體程序。1980年，RATLIFF 已經發展出的程式比JEP LONG 所寫的更爲有力。同年，RATLIFF開始在BYTE 雜誌的封底以VULCAN爲其程式作廣告，但比現在DBASE的功能還差得遠。1980年8月，RATLIFF 發表了一種VULCAN 革新版，其中附有索引及產生螢幕畫面特色，這個時候VULCAN 這項產品已超過了同類產品各對手的能力，但是銷售情況却相當緩慢，已賣出的VULCAN 不超過50個。

直至1980年的夏天，GEORGE TATE 開始聽聞到VU—LCAN的事蹟。TATE曾擔任主要軟件經銷商數年，他試行採用VULCAN ，也深受影响。當他知道這個軟體只賣出了5 套，他向RATLIFF 揚言他每個月可以賣50套。於是兩人很快地簽訂了一項保證予RATLIFF 豐厚專利費的合同。從此拋棄了VULCAN 的本名，而以dBASE II 來稱呼這項產品。正如第十二期介紹一樣，從來沒有dBASE I ，這只是一種行銷計劃，隱含著這是一個嶄新且改良過的版本，同時在文件說明中添加了一些特徵。

然後GEORGE TATE以DBASE II 和一個抽水唧筒（BILGE BUMP） 相比較的怪異廣告充斥於市場之上。資料庫的競爭對手們都不喜歡這個廣告，抽水唧筒的廠商也對此不滿意，但是這個廣告却大爲奏效。TATE另外成立了一家公司（ASHTON—TATE） 專門經銷這項產品。到了1982，ASHTON—TATE 一個月已售出超過2000套的dBASE II （每套US $ 495），RATLIFF 也由JPL 半退休了。

大家可以見到，DBASE II 是一個花了近十年光陰（1970年JEP LONG與RATLIFF 研究 JPLDIS開始）所寫成的高科技結晶品，（JPLDIS 是高度國防機密）來運作於微型電腦上，因此其它的庫存系統都在性能／功能上給比了下來。

基本系統需求

要應用DBASE II ，必須要有下列最基本的系統要求：一個CP／M—80系統，（使用Z 80或8080，8085作爲CPU）即Z 80 咭，80行顯示咭，64K 以上的主機記憶體（16K RAM咭）以存放CP／M系統。

DBASE II 第一步

要深入地認識DBASE 最好的方法是以一個實際的例子作爲引導，現在我們以PHONE LIST 作爲一個實例，它的格式如下：

NAME：20個字元長 （第 1 個欄）
ADDRESS 1：25個字元長 （第 2 個欄）
ADDRESS2 ：20個字元長 （第 3 個欄）
TEL ：10個字元長 （第 4 個欄）

現在首先鍵入 "CREATE"，DBASE II 會回應以：ENTER FILENAME，你可以輸入一個檔案名稱，長度在8個字元以內（這是CP／M 系統的限制，其中不能包含句點及空隔），讓我們叫它作NAMES。

當你這個時候按〔RETURN〕，dBASE II 便會自動建立一個名叫〔NAMES・DBF〕的檔案，在我們的例子之中，每一個記錄包括了四個欄，你可以寫入多個記錄到NAM—ES・DBF 之上，數量多少則視乎磁碟容量而定，由於D—BASE系統磁碟已佔據了大部份空間，只留下很小空間可供檔案寫入之用，因此最好你將檔案建立在B DRIVE 上，只要將一只CP／M 格式的磁碟放入B DRIVE 上，在點句標示之後鍵入SET DEFAULT TO B，dBASE便會將所有的資料存取交予B DRIVE處理。

由於現在我們要建立一個實用的PHONE LIST 資料庫，因此我們在按入CREATE之後，首先鍵入檔案名稱（NA—ME・DBF），之後便輸入欄名稱，類型，該欄的長度，小數點後的位數（見圖1所示）。其中欄名稱可以採用大／小楷輸入並最多可以達到10個字元長，不過開始及中間都不可以存有空格。類型（TYPE）是指資料種類，C 代表字符資料，而N是代表數字資料，L 代表邏輯運算（LOGICAL），以我們的PHONE LIST 爲例，所有欄上的資料都是字符。欄的濶度最多可以達到254個，但若果數字之中包括有小數時，便要緊記小數點需要佔據一個位長。

現在我們已大致明白dBASE 的檔案設立，每個記錄的結構定義，資料型態，我們鍵入例子後，螢幕應該如圖2中顯示，注意我們在005 該行上是按入RETURN的，這樣dB—ASE便會自動結束建立NAMES・DBF 這個檔案過程，而且會立即給予使用者輸入資料，這時若果你按入 "Y" 的話便可以輸入資料，各位不妨鍵入試試。但若果不預備在這個時候輸入的話你只要按N，便可以回到句點模式之中。

在任何時候你希望加添資料到檔案裏只要鍵入

USE NAMES
APPEND

圖(1)

```
CREATE
ENTER FILENAME:NAMES
ENTER RECORD STRUCTURE
AS FOLLOWS
 FIELD NAME, TYPE, WIDTH, DECIMAL
 001    PLACES.
```

```
.
. CREATE
ENTER FILENAME: NAMES
ENTER RECORD STRUCTURE AS FOLLOWS:
 FIELD    NAME,TYPE,WIDTH,DECIMAL PLACES
 001      NAME,C,20
 002      ADDRESS1,C,25
 003      ADDRESS2,C,20
 004      TEL,C,10
 005
INPUT DATA NOW? Y


RECORD # 00001
```

圖(2)

圖(3) 游標控制鍵簡介

CTRL Q —脫離系統，但不將記憶體內資料SAVE入碟
CTRL W —脫離系統，將所有記憶體內的資料SAVE入碟
CTRL E —將游標移返回上一欄
CTRL R —將記錄資料寫回磁碟並返回前一個記錄
CTRL T —刪除該欄，並將後隨的欄移上
CTRL Y —單獨刪除該欄，不影响其他欄
CTRL U —刪除記錄
CTRL S —將游標移退一個字元
CTRL D —將游標移前一個字元
CTRL G —刪除該字元
CTRL X —將游標移到下一個欄（或CTRL—F）
CTRL C —將記憶裏記錄寫回磁碟並移前一個記錄
CTRL V —加插或改寫一個字元
CTRL N —將欄移下一個位置，以便能加插一個新欄

便可以繼續你的輸入，在每一欄鍵入資料後按〔RETURN〕便可以到下一欄上繼續，在最後一欄按入〔RETURN〕便會顯示另一個記錄給你繼續輸入，若果你按入〔CTRL—Q〕的話便可以中止，回到點命令模式之中，這時你可以修改一些有錯的記錄，例如我們在第 2 個記錄錯誤輸入電話號碼，你可以鍵入EDIT 2 ，DBASE 便會把整個記錄顯示在螢幕上給你修輯，這時可以用圖 3 所列出的按鍵來移動CURSOR到任何欄，任何位置上修改內容，這時DBASE 所提供的改錯效能是全螢幕性的，有如文字處理器般去編輯資料。在完成修改後可以按〔CTRL—Q〕來回到點命令模式去。但各位要緊記在任何時間要離開DBASE 都要按入QUIT 。否則會失去重要資料。

資料輸出的方法

到了目前爲止我們都已經清楚了解到DBASE 在資料輸入的方法，現在我們要把所輸入的資料取回，DBASE 爲了儘可能把手續簡化，因此其內置程式只需使用三個命令（USE，LIST及DISPLAY） 便可將資料輸出，十分簡便。現在我們以PHONE LIST 爲例，逐步說明它的使用方法。

在點命令模式之中，首先我們鍵入USE NAMES ，現在若果你要顯示目前的記錄，只需按入DISPLAY ，若果要觀看全部內容，只要按LIST（你可以用CTRL—S 來暫停資料輸出，再次按入CTRL—S 可以使輸出繼續）。

DBASE 其中一項最強勁的功能是它的命令可以由使用者擴展與修改命令形式，使命令的輸入可以和高階語言（如BASIC等） 一樣加上語句在命令之後，使得效能可被擴展，它的格式如下：

LIST〔SCOPE〕〔OFF〕〔FOR<EXPRESSION>〕

其中SCOPE 是檔案上的資料欄名，如NAME，TEL等

OFF 會令檔案的記錄號碼不顯示出來

FOR 是用來連接尾隨條件敍述句的命令。

EXPRESSION 是條件敍述句，要充份掌握它的話，各位不妨參看以下例子作爲啓發。

USE NAME

LIST FOR NAME = "COM"，這個命令會使DBASE把NAME欄上起始字元爲COM的所有記錄列出，但記錄號碼亦同時出現。

LIST OFF FOR TEL < "8"

這個命令會使所有電話號碼少於 8 的一切記錄列出。

DISPLAY 命令的格式和LIST 一樣，基本上功能是相同的，閣下可以自己試一下。

以下我們把DBASE的認識來一個總括。首先DBASE是採用動詞形式的命令，在建立檔案時只要用簡易的CREA—TE命令，修改檔案內的記錄只要使用EDIT 命令，將資料列出可以用LIST及DISPLAY 命令，加添檔案資料時可以用APPEND 命令，到此我們不妨回顧一下各個使用過的同類型程序與DBASE 作比較；何者較好，但我要提醒各位，以上只不過是DBASE 在使用上的第一個層次入門——手動輸入命令操作，而DBASE 是可以採用程序來運作的，在以後有機會再和各位討論，因爲這是DBASE 的第二個使用層次。但無論怎樣，DBASE 內在潛能並非可以用三言兩語解釋清楚，希望我們一同能充實自己，再會！■

VISICALC處理超越函數及產生圖表的研究

陳耀南

在上期的SPREADSHEET專欄中我向各位介紹了SU—PERCALC 的優點及運作指令，使一些運行CP／M 或—MSDOS系統的朋友可以將擴頁運算技巧移到不同的操作系統之上。雖然因爲系統不同而不能將VISICALC 的資料交予SUPERCALC 運算，但技巧及基本原理其實是相同的，故此仍然保持了技術上的可移植性。

在今期筆者要爲各位介紹一些途徑令VISICALC 處理超越函數及產生圖表。假如閣下希望做有關科學或工程上的應用，這裏所討論將會滿足到你的需要。

我將在一個例子中，同時解說超越函數如@EXP ，@LN，和@SIN，與圖表格式／F＊。我們的目的是產生超越函數的圖表，因此必須在／F ＊格式的限制下做這件事，而它就是設計用來將行中的數字畫成簡單的長條圖。

首先我們清除SPREADSHEET以／CY 命令完成，再將方向指標改爲！。現在鍵入 1→3→6→20〔RET〕然後按＞A 1〔RET〕。

"星號"格式／F ＊顯示的星號個數與登錄值的整數部份相同。在A1鍵入／F＊，代以數字 1 的是一個星號（前面留一個空白）。

現在按→／F ＊。數字 3 代之以 3 個星號。繼續按→／F ＊→／F ＊。A3 顯示 6 個星號，而A4顯示 8 個（9 個字元行寬的最大數）。鍵入／GC12〔RET〕且注意A4 現在顯示11個星號。用／CY清除工作紙。

爲了將一個函數用圖表示，我們必須先提供一串引數的值，且計算每一個引數值的函數值。爲簡單起見，我們先用一個線性函數，如f （X）＝2.5＊X。鍵入如下：

```
＞A20〔RET〕
.1→.1〔RET〕
＞A1：（S）
 ＋A20  →＋←＋B20〔RET〕
／R：→.A17：RN
```

在第20列我們爲引數定義了一個起始值和間隔值。然後用公式和複製指令計算引數值從.1到1.7 （在A1到A17 的位置）。然後我們將計算出相對應的線性函數值。鍵入如下：

```
 ＞B1〔RET〕
 2.5＊A1
 ／R：→.B17：R
```

函數值爲.25，.5，.75，……，到4.25。爲了看這些在星號格式中的情形，我們將再一次設定格式和複製。

鍵入／F ＊／R ：→B17：R 。這將會是甚麼圖形呢？它或許不是你心中想要的圖形。問題是函數值並不是正好落在1, 2, 3, 等使到可以產生一個，二個，或三個星號的位置。

讓我們用／FD／R：→.B17：R回到數字的顯示。

我們如何產生一個較好的星號格式顯示？首先我們將設定一個較寬的行寬使我們能用較好的解析度去表示較寬範圍的函數值。這使我們能產生17個星號，然後我們將"刻度"（SCALING） 這些函數值將.25 到4.25 放到0到17的範圍內。鍵入如下所示：

```
＞B 18〔RET〕
／－－－→@MAX （B1．B17）〔RET〕
／GC 8〔RET〕
＞C1〔RET〕
／WV；／GC 18〔RET〕
17／B 19＊@ABS （B1）〔RET〕
```

在B 19，我們用@MAX 函數去找出在此範圍內的函數值的上限。爲簡單起見，我們用函數值的絕對值，所以下限爲零。你也可以同時用@MAX 和@MIN去刻度。我們也已在窗幕的右邊給星號設定較寬的行寬。

在C1的公式是用來乘以每個函數值，如B1 乘17（最大的星號數目）再除以B 19（最大函數值）。此公式的值將在0 到17／B 19＊B 19即17之間。

現在鍵入／F＊以設定C1的顯示格式。鍵入／R ：→.C 17：NR ，這時你注意螢幕的變化。

你的螢幕將如圖 1 所示出現一條合理且近似的直線。另外，如果已做好我們的工作，我們可以畫出任何B行中的函數值，而不只是線性函數f （X）＝2.5＊X。我們將調整引數範圍使我們能畫出函數值中較有興趣的一段。現在你可以鍵入以下：

```
；＞A 20：（S）
1→.25〔RET〕
＞B 1〔RET〕
@LN （A1）〔RET〕
／R：B2．B17：R
```

注意這時在B 19，@MAX （B1……B17）的值已變爲1.6 0944。這個值會影響在第C 行中的公式，使得結果仍然在 0 到17的範圍內。如果你對自然對數的 圖並 不十分熟悉，將你的頭轉一下，且將列想像成X 軸，而行爲Y 軸。（參看圖 2）

```
 .1          .25 #
 .2           .5 ##
 .3          .75 ###
 .4            1 ####
 .5         1.25 #####
 .6          1.5 ######
 .7         1.75 #######
 .8            2 ########
 .9         2.25 #########
  1          2.5 ##########
1.1         2.75 ###########
1.2            3 ############
1.3         3.25 #############
1.4          3.5 ##############
1.5         3.75 ###############
1.6            4 ################
1.7         4.25 #################
----------------------
            4.25
 .1           .1
```

圖(1)

```
               0
   1   .223143551296 ##
1.25   .405465108211 ####
 1.5   .559615787968 #####
1.75   .693147180644 ######
   2   .810930216192 #######
2.25   .91629073194 ########
 2.5   1.0116009118 #########
2.75   1.0986122886 ##########
   3   1.1786549961 ###########
3.25   1.2527629683 ############
 3.5   1.3217558399 #############
3.75   1.386294361 ##############
   4   1.4469189828 ##############
4.25   1.5040773966 ###############
 4.5   1.5581446179 ###############
4.75   1.6094379123 ################
   5
----------------
       1.6094379123
                .25
```

圖(2)

```
   1     2.718281828
1.25     3.4903429569
 1.5     4.481689069
1.75     5.7546026751
   2     7.3890560864
2.25     9.4877357647 #
 2.5     12.182493957 #
2.75     15.642631882 #
   3     20.085536919 ##
3.25     25.790339911 ##
 3.5     33.115451952 ###
3.75     42.521081985 ####
   4     54.598150011 #####
4.25     70.10541225 #######
 4.5     90.017130784 #########
4.75     115.58428451 ############
   5     148.41315909 ###############
-----------------
         148.41315909
                  .25
```

圖(3)

現在讓我們試試指數函數。鍵入

@EXP (A1) [RET]

／R：B2. B17：R

最大的函數值現爲148.43 ，如圖３所示，而且每一個值均已在第C行中刻度在０到17的範圍內。這個圖與你期望的相近嗎？最後，我們將畫正弦函數。由於三角的計算是用RADIAN（徑度），我們將用@SIN（@PI＊A1）和一個不同的引數範圍以使我們能在第C行中得到整個正弦圖形。

鍵入如下：

＞A 20 [RET]

:03→.06 [RET]

＞B1 [RET]

@SIN (@PI＊A1) [RET]

／R：B2. B17：R

如果進行順利，你將在第C行中有一個合理的正弦數圖形出現，就如下面的圖４一般。

這時你或許希望用不同的引數值和函數去做些實驗，爲了方便讀者，以上的４個圖形程式都會加入到今期的程式磁碟中，檔案名稱分別爲：

圖１的： VC. LINEAR

圖２的： VC. LN

圖３的： VC. EXP

圖４的： VC. SIN

```
.03   .094108313324 #
.09   .278991106051 ###
.15   .453990499742 ######
.21   .612907053686 #########
.27   .750111069695 ###########
.33   .860742027011 ############
.39   .940880769035 ##############
.45   .987688340628 ##############
.51   .99950656047 ##############
.57   .975916762066 ##############
.63   .917754625801 ##############
.69   .827080574373 ##############
.75   .707106781249 ###########
.81   .562083377877 ########
.87   .397147890679 #####
.93   .21814324145 ###
.99   .031410759142
----------------
       .99950656047
                .06
```

圖(4)

利用電腦編寫日文書信或通告

日文文字處理系統

胡永達

DEAR READER:
こんにちは。おけんいでか。どうぞおな
まえは。わたしは ルイです。どうぞ よろし
く。
I PRODUCT TWO FONT
1.ひらがな
2.かたかな
DO YOU LIKE TRESE JAP
ANESE LANGUAGE?

日本的浪潮越來越吹近香港,而學日文也十分普遍。很多青少年都去學習日本的文化—日文。慣得日文的讀者，是否感到你的電腦只能夠顯示英文實在十分之單調，就算花數百元去買張"中文咭"也只得中、英文的顯示。所以筆者設計了兩套的日文字來方便大家。

這兩套日文字不但可以加入圖畫上，而且還可以作書信的來往。它是用SHARP TABLE 來畫出來，所以讀者們可以用DRAW或XDRAW的指令來加入自己的程序內，不過要首先載入筆者所設計之日文字體。字體一共分兩套：

1.平仮名（ひらがな）
HIRAGANNA

2.片仮名（かたかな）
KATAKANA

另外筆者亦設計了一套英文字包括各類符號及小楷字。用法：

首先載入程序（JAPANESE WRITER），然後載入字體的圖形。磁碟內已存有三套字體，兩套日文及一套英文。日文爲〕HIRAGANA及〕KATAKANA，英文有〕PENM—EN 。如果只需要在空白畫面內鍵入日文，首先按"X"，"Y"便可。因爲印字機的不同牌子有不同指令，所以讀者們自行加入指令在程式內。每鍵一個制便會畫出一個日文，這可以減少利用HPLOT 之麻煩。所有日文字可以在表(一)

表(一) 平仮名 〕HIRAGANA

I	201	は	D	196	な	:	186	た	0	176	さ	&	166	か	!	161	あ
L	204	ひ	E	197	に	<	188	ち	2	178	し	'	168	き	"	162	い
O	207	ふ	F	198	ぬ	>	190	つ	4	180	す	*	170	く	#	163	う
R	210	へ	G	199	ね	?	192	て	6	182	せ	+	172	け	$	164	え
U	213	ほ	H	200	の	@	194	と	8	184	そ	.	174	こ	%	165	お
,	167	が	g	231	ん	e	229	わ	;	224	ら	1	221	や	X	216	ま
)	169	ぎ							a	225	り				Y	217	み
+	171	ぐ							b	226	る	↑	222	ゆ	Z	218	む
-	173	げ							c	227	れ				[	219	め
/	175	ご				f	230	を	d	228	ろ	_	223	よ	n	238	も
m	237	っ	h	232	。	R	203	ば	J	202	ば	:	187	だ	1	177	ざ
			i	233	、	S	206	び	M	205	び	=	189	ぢ	3	179	じ
			j	234	ゃ	Q	209	ぶ	P	208	ぶ	?	191	づ	5	181	ず
			k	235	ゅ	T	212	べ	S	211	べ	A	193	で	7	183	ぜ
			l	236	ょ	W	215	ぼ	V	214	ぼ	C	195	ど	9	185	そ

及表(二)查閱。

如果讀者們利用本程序，大可以在朋友面前誇耀自己的日文及電腦知識。

指令鍵有：

本程序（JAPANESE WRITER）的控制指令鍵如下：

(1)，(2)，(3)——載入預先設計好之字體。
(X)——清洗畫面。
(L)——載入預先設好之圖畫。
(S)——錄入已加上日文之圖畫及書信入磁碟上。
(C)——讀取碟內之CATALOG。
(Q)——離開本程序。
(RTN)——開始鍵入日文。
(CTRL－R)－返回MENU
(CTRL－A)－游標移向上（每格）
(CTRL－B)－清洗字體（一點）
(CTRL－E)－清洗字體（全字）
(CTRL－F)－轉變字體（1－3）
(CTRL－G)－將全畫面分爲16格
(CTRL－H)－游標向左移（每格）
(CTRL－K)－游標向左移（每點格）
(CTRL－L)－游標向右移（每點格）
(CTRL－M)－開新行
(CTRL－O)－選擇顏色（0－6）
(CTRL－S)－游標向上移（每點格）
(CTRL－U)－游標向右移（每格）
(CTRL－W)－將全行字體及圖畫清洗
(CTRL－X)－游標向下移（每點格）
(CTRL－Z)－游標向下移（每格）

本文共由五個程式組成，包括如下：

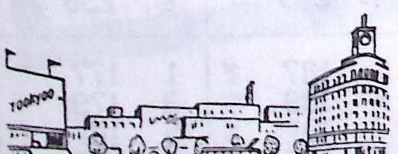
```
JAPANESE WRITER
CURSORS
JHIRAGANA.F
 JKATAKANA.F
JPENMAN
```

其中第一個爲主程式，用來操縱繪字第二個爲一控制游標的副程式。第三至第五是字型資料，包括有二套日文字及一套英文／數目字。

鑑於五個程式所佔用的篇幅會很長，我們決定只將這五個程式收入本期的程式磁碟上，而不在雜誌刊出，希望讀者體諒。本期程式磁碟可於各特約經銷處購得，或用郵購方式。■

表(二) 片仮名

															按鍵	ASCII字	日文字
I	201	ハ	D	196	ナ	:	186	タ	0	176	サ	&	166	カ	!	161	ア
L	204	ヒ	E	197	ニ	←	188	チ	2	178	シ	(	168	キ	"	162	イ
O	207	フ	F	198	ヌ	→	190	ツ	4	180	ス	*	170	ク	#	163	ウ
R	210	ヘ	G	199	ネ	↓	192	テ	6	182	セ	,	172	ケ	$	164	エ
U	213	ホ	H	200	ノ	B	194	ト	8	184	ソ	.	174	コ	%	165	オ
			f	230	ン	¢	229	ワ	`	224	ラ	]	221	ヤ	X	216	マ
									a	225	リ				y	217	ミ
									b	226	ル	↑	222	ユ	Z	218	ム
									c	227	レ				[	219	メ
						g	231	ヲ	d	228	ロ	_	223	ヨ	s	243	モ
i	233	・	K	203	パ	J	202	バ	;	187	ダ	1	177	ザ	'	168	ガ
h	232	、	N	206	ピ	M	205	ビ	=	189	ヂ	3	179	ジ	)	169	ギ
r	242	ャ	Q	209	プ	P	208	ブ	?	191	ヅ	5	181	ズ	+	171	グ
j	234	ュ	T	212	ペ	S	211	ベ	@	193	デ	7	183	ゼ	-	173	ゲ
k	235	ョ	W	215	ポ	V	214	ボ	C	195	ド	9	185	ゾ	/	175	ゴ
												q	241	ォ	l	236	ッ
															m	237	ァ
															n	238	ィ
															o	239	ゥ
															p	240	ェ



瞬息間便可更改任何字串及變數

執行速度異乎想像的——

APPLESOFT 全域編輯器

各位程序員在編寫程式時，最令人頭痛的一項事項就是要更改整個程式中的變數或字串，偶一不愼很容易便將要更改的變數或字串遺漏，前功盡廢。

不過，看完本文及使用過本文所述的APPLESOFT 全域性編輯器之後，閣下，可以在萬分之一秒的時間內找出要更改的字串／變數，並由電腦自己替你改變該字串／變數。

APPLESOFT 全域性編輯器程式佔用了主記憶體上的2.7K 記憶位置，適合48K系統使用。

當我們執行本程式後，它會放在DOS 之下的位址，並自動設定HIMEM 來保護程式內容不被其他程式所覆蓋。大家按入＆後，程式就會即時進入編輯狀態，並要求用家輸入要尋找的字串或變數（當然，你在事前經已載入要編輯的一APPLESOFT 程式在主記憶上），然後再打入代替該字串／變數的字串，字串最長可以有32個字母。假如中途打錯字，可用←鍵來更正。然後你便會看到一個指令表。指令表包括下列各項功能：

(1) BEGIN
(2) SUPRESS LIST
(3) AUTO REPLACE
(4) LINE RANGE
(5) END

大家只要按下B）EGIN ，編輯器會立即進行工作。按E）ND終止編輯工作，返回BASIC。

若然按下A）UTO REPLACE ，則編輯器會自動將待修改的程式內所有要改的變數全部改動過來，使用者無須費神，亦因此原故，要非常小心使用此指令。

通常情況下，編輯器在執行工作時，會將待修正的程式LIST 出來，但假如閣下無興趣看著程式一邊LIST 出來，一邊改動，則可用S）UPPRESS LIST 指令來進行，這指令會連同A）UTO指令一併執行工作。

最後一個指令L）INE RANGE 則是容許閣下設定一個行數範圍，編輯器會自動在這行句範圍內找出要修改的東西。

當任何一個指令（L、S、A）被選擇後，這些指令會以反白的形式顯示，以表示目前正處於的狀態。若要取消這些指令，再按一次該指令即可。

除了A及S 指令外，編輯器會在執行工作時，被修改的程式會在螢幕上顯示，曾遇到要修改的字串時，該字串會以反白顯示，此時按Y會將該字串更改，按E會返回BASIC，其他任何鍵會讓程式繼續顯示。相信大家對於這個全域性編輯器的使用方法清楚明白了。

在此不妨舉一個實例，假定有下列一個APPLE SOFT 程式其中的HELLO 改動爲GREETING 。

```
10  REM HELLO
20  PRINT CHR $(4); "CATALOG"
30  END
```

按下＆鍵，以便進入編輯器中，打入HELLO 字串，告訴電腦要更改，然後打入代替HE LLO的GREETING ，再按B。編輯器首先會將程式LIST 出，找到行句10的HELLO字串後，會以反白模式顯示該字串，此時按Y便可將字串改爲GREETING，然後程式會繼續。

無論任何時刻，閣下都可按＆來修改任何APPLESOFT 程式。由於編輯器是以＆向量來啓動，假如閣下有需要用到＆向量作其他用途，則只要CALL 35712 ，依然可以觸動編輯器。

本程式是利用LISA 2.5 ASSEMBLER 寫成，擁有一LISA 2.5 的讀者可依列表打入，至於沒有LISA 2.5 或不懂ASSEMBLER 的讀者，可以按左手邊的機械碼輸入，並用下列指令存入磁碟：

BSAVE GLOBAL EDITOR，A$8B00，L$A7E。

讀者可在本期程式磁碟中找到一個經ASSEMBLER 過的檔案GLOBAL EDITOR。

假如有讀者需要此程式的SOURCE CODE，可向讀者服務部查詢。 ■

APPLESOFT全域編輯器

```
]ASM
**END OF PASS 1

**END OF PASS 2

0800                    1  ; GLOBAL EDITOR
0800                    2  ; BY COMPUHOUSE
0800                    3  ; COPYRIGHT (C) 1984
0800                    4  ;
0800                    5  ;
0800                    6  ; LISA ASSEMBLER
0800                    7  ;
0800                    8  ;
8B00                    9          ORG $8B00
8B00                   10          OBJ $800
FC9C                   11  CLEOL   EQU $FC9C
DD7B                   12  FRMEVL  EQU $DD7B
E752                   13  GETADR  EQU $E752
FD8E                   14  CROUT   EQU $FD8E
FD0C                   15  RDKEY   EQU $FD0C
FDED                   16  COUT    EQU $FDED
FD6A                   17  GETLN   EQU $FD6A
FF3A                   18  BELL    EQU $FF3A
0060                   19  BL      EQU $60          ;BLANK - CONVERTED
0007                   20  SEVEN   EQU $7
000D                   21  CR      EQU $0D          ;CARR RET - CON
0088                   22  BA      EQU $88          ;BACK ARROW
00EA                   23  LT      EQU $EA          ;LAST TOKEN VALUE
0050                   24  LINUM   EPZ $50
0024                   25  CH      EPZ $24          ;HORIZ POS
0025                   26  CV      EPZ $25          ;VERT POS
0032                   27  INVFLG  EPZ $32
0022                   28  WNDTOP  EPZ $22
0023                   29  WNDBTM  EQU $23
0073                   30  HIMEM   EPZ $73
006F                   31  FRETOP  EPZ $6F          ;STRING STORAGE
00FD                   32  PNTADD  EPZ $FD          ;PRINTER POINTER
00FD                   33  TPTR    EPZ $FD          ;TABLE POINTER
00B8                   34  TXTPTR  EPZ $B8          ;TEXT POINTER
00AF                   35  PGEND   EPZ $AF          ;PROGRAM END
0069                   36  VARTAB  EPZ $69          ;VARIABLE PNTR
006B                   37  ARYTAB  EPZ $6B          ;ARRAY PNTR
006D                   38  STREND  EPZ $6D          ;END OF STORAGE
FC58                   39  HOME    EQU $FC58
FB5B                   40  TABV    EQU $FB5B
03F6                   41  AMPERV  EQU $3F6
8B00                   42  *
8B00                   43  *  INITIALIZE GE
8B00                   44  *
8B00 A9 13             45  SETUP   LDA #START       ;PROTECT HIMEM
8B02 85 73             46          STA HIMEM
8B04 85 6F             47          STA FRETOP
8B06 8D F6 03          48          STA AMPERV
8B09 A9 8B             49          LDA /START
8B0B 85 74             50          STA HIMEM+1
8B0D 85 70             51          STA FRETOP+1
8B0F 8D F7 03          52          STA AMPERV+1
8B12 60                53          RTS
8B13                   54  *
8B13                   55  *  START MAIN PROG
8B13                   56  *
8B13 20 58 FC          57  START   JSR HOME        ;CLR SCREEN
8B16 A9 00             58          LDA #0           ;SET CUR POS
8B18 85 24             59          STA CH
8B1A A9 11             60          LDA #17
8B1C 20 5B FB          61          JSR TABV
8B1F A9 07             62          LDA #FINDF       ;MENU
8B21 85 FD             63          STA PNTADD
8B23 A9 8C             64          LDA /FINDF
8B25 85 FE             65          STA PNTADD+1
8B27 20 78 8B          66          JSR PRINT
8B2A A0 00             67          LDY #0
8B2C 8C D6 8D          68          STY TOFF
8B2F 20 48 8B          69          JSR STCHAR       ;READ INPUT LINE
8B32 A9 13             70          LDA #REPF        ;REPLACE STRING
8B34 85 FD             71          STA PNTADD
8B36 A9 8C             72          LDA /REPF
8B38 85 FE             73          STA PNTADD+1
8B3A 20 78 8B          74          JSR PRINT
8B3D A0 20             75          LDY #$20         ;BUFF OFFSET
8B3F 8C D6 8D          76          STY TOFF
8B42 20 48 8B          77          JSR STCHAR       ;READ LINE
8B45 4C 65 8C          78          JMP MAIN         ;NEXT PART
8B48 A2 FF             79  STCHAR  LDX #$FF
8B4A E8                80  ST2     INX
8B4B 20 0C FD          81  ST25    JSR RDKEY
8B4E 20 ED FD          82          JSR COUT
8B51 C9 88             83          CMP #BA          ;BACK ARROW?
8B53 D0 13             84          BNE ST3          ;NO
8B55 CE D6 8D          85          DEC TOFF
8B58 CA                86          DEX
8B59 E0 FF             87          CPX #$FF         ;BEGINNING OF LINE?
8B5B D0 EE             88          BNE ST25
8B5D EE D6 8D          89          INC TOFF
8B60 E6 24             90          INC CH
8B62 20 3A FF          91          JSR BELL
8B65 4C 4A 8B          92          JMP ST2
8B68 29 7F             93  ST3     AND #$7F         ;CONVERT ASCII
8B6A AC D6 8D          94          LDY TOFF
8B6D 99 87 8B          95          STA BUFF1,Y      ;PUT CHAR IN BUFFER
8B70 EE D6 8D          96          INC TOFF
8B73 C9 0D             97          CMP #CR
8B75 D0 D3             98          BNE ST2
8B77 60                99          RTS
8B78                  100  *
8B78                  101  *  PRINT SUBR.
8B78                  102  *
8B78 A0 00            103  PRINT   LDY #0
8B7A B1 FD            104          LDA (PNTADD),Y
8B7C AA               105          TAX
8B7D C8               106  PRI     INY
8B7E B1 FD            107          LDA (PNTADD),Y
8B80 20 ED FD         108          JSR COUT         ;PRINT IT
8B83 CA               109          DEX
8B84 D0 F7            110          BNE PRI          ;GET NEXT CHAR
8B86 60               111          RTS
8B87                  112  *
8B87                  113  *  STORAGE AREA
8B87                  114  *
8B87                  115  BUFF1   DFS $20          ;FIND STRING BUFFER (ACTUAL)
8BA7                  116  BUFF2   DFS $20          ;REPLACE STRING BUFFER (ACT)
8BC7                  117  BUFF3   DFS $20          ;FIND STRING (TOKENIZED)
8BE7                  118  BUFF4   DFS $20          ;REPL STRING (TOKENIZED)
8C07                  119  *  STRINGS
8C07 0B D3 C5         120  FINDF   STR "SEARCH FOR "
8C0A C1 D2 C3
8C0D C8 A0 C6
8C10 CF D2 A0
8C13 0B D2 C5         121  REPF    STR "REPLC WITH "
8C16 D0 CC C3
8C19 A0 D7 C9
8C1C D4 C8 A0
8C1F 45               122  OPTBL   HEX 45
8C20 0D               123          HEX 0D
8C21 A0 A0 A0         124          ASC "       "
8C24 A0 A0 A0
8C27 A0
8C28 BC C2 BE         125          ASC "<B>EGIN     <S>UPP LST  <E>ND"
8C2B C5 C7 C9
8C2E CE A0 A0
8C31 A0 A0 A0
8C34 BC D3 BE
8C37 D5 D0 D0
8C3A A0 CC D3
8C3D D4 A0 A0
8C40 BC C5 BE
8C43 CE C4
8C45 0D               126          HEX 0D
8C46 C3 CD CD         127          ASC "CMMD   <A>UTO RPL  <L>INE RANGE"
8C49 C4 A0 A0
8C4C A0 BC C1
8C4F BE D5 D4
8C52 CF A0 D2
8C55 D0 CC A0
8C58 A0 BC CC
8C5B BE C9 CE
8C5E C5 A0 D2
8C61 C1 CE C7
8C64 C5
8C65                  120  *
8C65                  129  *  MAIN RTN - TOKENIZES
8C65                  130  *
8C65 A9 40            131  MAIN    LDA #$40
8C67 8D D6 8D         132          STA TOFF         ;PRESET OFFSET
8C6A A2 00            133          LDX #$0          ;PRESET BUFF OFFSET
8C6C 20 AE 8C         134          JSR TOKEN
8C6F A2 20            135          LDX #$20         ;PRSET BUFF OFFSET
8C71 A9 60            136          LDA #$60
8C73 8D D6 8D         137          STA TOFF         ;PRESET TBUFF OFFSET
8C76 20 AE 8C         138          JSR TOKEN
8C79 A2 00            139          LDX #0           ;NOW SET COUNTERS
8C7B 8E D6 8D         140          STX TOFF         ;USE TOFF AS A COUNTER
8C7E 8E D7 8D         141          STX TOFF1        ;PRESET TO ZERO
8C81 A0 00            142  M0      LDY #0
8C83 BD 87 8B         143  M1      LDA BUFF1,X
8C86 F0 05            144          BEQ M3           ;DONE
8C88 E8               145          INX
8C89 C8               146          INY
8C8A 4C 83 8C         147          JMP M1
8C8D 98               148  M3      TYA
8C8E AE D6 8D         149          LDX TOFF
8C91 9D D8 8D         150          STA BCNT,X
8C94 AD D6 8D         151          LDA TOFF
8C97 C9 03            152          CMP #3           ;DONE?
8C99 F0 10            153          BEQ M4
8C9B EE D6 8D         154          INC TOFF         ;INC OFFSET
8C9E AD D7 8D         155          LDA TOFF1
8CA1 18               156          CLC
8CA2 69 20            157          ADC #$20
8CA4 8D D7 8D         158          STA TOFF1
8CA7 AA               159          TAX              ;UPDATE BUFF OFFSET
8CA8 4C 81 8C         160          JMP M0           ;GO BACK
8CAB 4C 12 90         161  M4      JMP OPTIONS      ;PRINT SUB MENU
8CAE                  162  *
8CAE                  163  *
8CAE                  164  *  END OF TOKENIZATION
8CAE                  165  *
8CAE                  166  *
8CAE                  167  *  TOKEN ROUTINE - ENTRY
8CAE                  168  *
8CAE BD 87 8B         169  TOKEN   LDA BUFF1,X      ;GET BUFFER CHAR
8CB1 C9 22            170          CMP #$22         ;"SIGN
8CB3 F0 76            171          BEQ QUOTE
8CB5 C9 20            172          CMP #$20         ;ODD BLANK
8CB7 D0 04            173          BNE TK1          ;NOT A SPACE
8CB9 E8               174          INX              ;GET NEXT CHAR
8CBA 4C AE 8C         175          JMP TOKEN
8CBD C9 0D            176  TK1     CMP #CR          ;CARR RET?
8CBF D0 0C            177          BNE TK2
8CC1 A9 00            178  TK15    LDA #0           ;DELIMIT LINE WITH ZERO
8CC3 9D 87 8B         179          STA BUFF1,X
```

```
8CC6 AE D6 8D   180          LDX TOFF
8CC9 9D 87 8B   181          STA BUFF1,X
8CCC 60         182          RTS                 ;DONE
8CCD 20 65 8D   183  TK2     JSR TSEARCH         ;TOKENIZE
8CD0 AD D4 8D   184          LDA TKSW            ;TOKEN SWITCH SET?
8CD3 D0 12      185          BNE TK4             ;TOKEN FOUND
8CD5 AC D6 8D   186          LDY TOFF            ;LOAD TBUFF OFFSET
8CD8 BD 87 8B   187          LDA BUFF1,X         ;GET CHAR
8CDB 29 7F      188          AND #$7F            ;CONVERT ASCII
8CDD 99 87 8B   189          STA BUFF1,Y
8CE0 EE D6 8D   190          INC TOFF
8CE3 E8         191          INX                 ;INC BUFFER
8CE4 4C AE 8C   192          JMP TOKEN
8CE7 AC D6 8D   193  TK4     LDY TOFF            ;LOAD TBUFF OFFSET
8CEA AD D5 8D   194          LDA TKCNTR          ;GET TOKEN VALUE
8CED C9 C5      195          CMP #$C5            ;AT TOKEN
8CEF D0 18      196          BNE TK40            ;NOT AT TOKEN
8CF1 48         197          PHA                 ;SAVE CHAR
8CF2 E8         198          INX
8CF3 E8         199          INX                 ;CHK FOR 3RD CHAR N?
8CF4 BD 87 8B   200          LDA BUFF1,X         ;GET CHAR
8CF7 C9 4E      201          CMP #$4E            ;N?
8CF9 D0 0B      202          BNE TK43            ;NOT ATN TOKEN
8CFB 68         203          PLA
8CFC A9 E1      204          LDA #$E1            ;ATN TOKEN
8CFE CA         205          DEX
8CFF CA         206          DEX                 ;RESTORE BUFFER CNTR
8D00 EE D3 8D   207          INC CHCNT           ;TOK LEN
8D03 4C 09 8D   208          JMP TK40            ;STORE ATN TOKEN
8D06 CA         209  TK43    DEX                 ;REM - NOT ATN
8D07 CA         210          DEX                 ;RESTORE BUFF CNTR
8D08 68         211          PLA                 ;RESTORE STACK
8D09 99 87 8B   212  TK40    STA BUFF1,Y
8D0C EE D6 8D   213          INC TOFF            ;INC TOKEN BUFF OFFSET
8D0F C9 B2      214          CMP #$B2            ;REM TOKEN
8D11 D0 0E      215          BNE TK44
8D13 A0 00      216  TK41    LDY #0              ;FOR REM
8D15 8C DC 8D   217  TK42    STY SV              ;SWITCH VAR
8D18 E8         218          INX
8D19 E8         219          INX                 ;INC BUFFER
8D1A E8         220          INX
8D1B AC D6 8D   221          LDY TOFF            ;GET REP OFFSET
8D1E 4C 39 8D   222          JMP Q2
8D21 C9 83      223  TK44    CMP #$83            ;DATA TOKEN?
8D23 D0 34      224          BNE TK45
8D25 E8         225          INX                 ;CORRECT FOR DATA
8D26 A0 3A      226          LDY #$3A            ;COLON
8D28 4C 15 8D   227          JMP TK42
8D2B 8D DC 8D   228  QUOTE   STA SV              ;SAVE " SIGN
8D2E AC D6 8D   229  Q0      LDY TOFF            ;BUFF OFFSET
8D31 99 87 8B   230          STA BUFF1,Y
8D34 E8         231  Q1      INX
8D35 C8         232          INY
8D36 EE D6 8D   233          INC TOFF            ;INC CNTRS
8D39 BD 87 8B   234  Q2      LDA BUFF1,X         ;GET CHAR
8D3C CD DC 8D   235          CMP SV
8D3F D0 0B      236          BNE Q3
8D41 99 87 8B   237          STA BUFF1,Y
8D44 E8         238          INX                 ;INC CNTRS
8D45 C8         239          INY
8D46 EE D6 8D   240          INC TOFF
8D49 4C AE 8C   241          JMP TOKEN
8D4C C9 0D      242  Q3      CMP #CR             ;CARR RET?
8D4E D0 03      243          BNE Q4
8D50 4C C1 8C   244          JMP TK15
8D53 99 87 8B   245  Q4      STA BUFF1,Y         ;STORE CHAR
8D56 4C 34 8D   246          JMP Q1
8D59 AC D3 8D   247  TK45    LDY CHCNT           ;LEN OF TOKEN
8D5C E8         248  TK5     INX
8D5D 88         249          DEY                 ;DEC CNTR
8D5E C0 00      250          CPY #0              ;ZERO?
8D60 D0 FA      251          BNE TK5
8D62 4C AE 8C   252          JMP TOKEN
8D65            253  *
8D65            254  *  TSEARCH - TOKENIZE
8D65            255  *       ON ENTRY X = OFFSET IN BUFFER
8D65            256  *       ON EXIT X RESTORED, TKSW = 1 IF TOKEN FOUND
8D65            257  *            AND CHCNT = TOKEN LENGTH
8D65            258  *
8D65 A9 80      259  TSEARCH  LDA #$80            ;FIRST TOKEN VALUE
8D67 8D D5 8D   260          STA TKCNTR
8D6A 8E D1 8D   261          STX TEMPX           ;SAVE X
8D6D A9 00      262          LDA #0
8D6F 8D D3 8D   263          STA CHCNT           ;PRESET CNTRS
8D72 8D D4 8D   264          STA TKSW
8D75 A8         265          TAY                 ;PRESET Y REG
8D76 A9 84      266          LDA #TABLE          ;SET TABLE POINTER
8D78 85 FD      267          STA TPTR
8D7A A9 93      268          LDA /TABLE
8D7C 85 FE      269          STA TPTR+1
8D7E BD 87 8B   270  TS1     LDA BUFF1,X
8D81 D1 FD      271          CMP (TPTR),Y        ;CHECK FIRST CHAR W/TABLE
8D83 F0 1D      272          BEQ TS5             ;MATCHES
8D85 20 C7 8D   273  TS3     JSR TINC
8D88 B1 FD      274          LDA (TPTR),Y
8D8A C9 60      275          CMP #BL
8D8C D0 F7      276          BNE TS3             ;NOT DONE
8D8E 20 C7 8D   277          JSR TINC            ;INC TOKEN PTR
8D91 EE D5 8D   278          INC TKCNTR          ;INC TOKEN CNTR
8D94 AD D5 8D   279          LDA TKCNTR
8D97 C9 EA      280          CMP #LT             ;LAST TOKEN?
8D99 D0 E3      281          BNE TS1
8D9B AE D1 8D   282  TS4     LDX TEMPX           ;RESTORE X
8D9E EE D3 8D   283          INC CHCNT
8DA1 60         284          RTS

8DA2 E8         285  TS5     INX
8DA3 20 C7 8D   286          JSR TINC
8DA6 B1 FD      287          LDA (TPTR),Y
8DA8 C9 60      288          CMP #BL             ;BLANK?
8DAA D0 06      289          BNE TS6             ;NOT DONE
8DAC EE D4 8D   290          INC TKSW            ;SET TOKEN SWITCH
8DAF 4C 9B 8D   291          JMP TS4             ;DONE
8DB2 EE D3 8D   292  TS6     INC CHCNT           ;TOKEN CHAR CNTR
8DB5 BD 87 8B   293          LDA BUFF1,X         ;GET CHAR
8DB8 D1 FD      294  TS7     CMP (TPTR),Y
8DBA F0 E6      295          BEQ TS5             ;MATCHES
8DBC AE D1 8D   296          LDX TEMPX           ;RESTORE X
8DBF A9 00      297          LDA #0              ;RESET CHAR CNTR
8DC1 8D D3 8D   298          STA CHCNT
8DC4 4C 85 8D   299          JMP TS3
8DC7 E6 FD      300  TINC    INC TPTR            ;INC TABLE POINTER
8DC9 A5 FD      301          LDA TPTR
8DCB D0 02      302          BNE TI1
8DCD E6 FE      303          INC TPTR+1
8DCF 60         304  TI1     RTS
8DD0            305  *
8DD0            306  *
8DD0            307  *  VARIABLES
8DD0            308  *
8DD0            309  TEMPA   DFS $1
8DD1            310  TEMPX   DFS $1
8DD2            311  TEMPY   DFS $1
8DD3            312  CHCNT   DFS $1
8DD4            313  TKSW    DFS $1              ;SWITCH
8DD5            314  TKCNTR  DFS $1              ;TOKEN CNTR
8DD6            315  TOFF    DFS $1              ;TOKEN BUFF OFFSET
8DD7            316  TOFF1   DFS $1              ;OFFSET COUNTER
8DD8            317  BCNT    DFS $4              ;LEN CNTERS OF BUFFERS
8DDC            318  SV      DFS $1
8DDD            319  *  SAVE CUR POS
8DDD            320  *       ALL REGS INTACT
8DDD 48         321  SVCUR   PHA                 ;SAVE ACC
8DDE A5 24      322          LDA CH
8DE0 8D EA 8D   323          STA CHT
8DE3 A5 25      324          LDA CV
8DE5 8D EB 8D   325          STA CVT
8DE8 68         326          PLA                 ;RESTORE ACC
8DE9 60         327          RTS
8DEA            328  CHT     DFS $1              ;STORAGE
8DEB            329  CVT     DFS $1
8DEC            330  *  RESTORE CUR POS
8DEC            331  *       ALL REGS INTACT
8DEC 48         332  RECUR   PHA                 ;SAVE ACC
8DED A5 24      333          LDA CH              ;RESET CVT
8DEF CD EA 8D   334          CMP CHT
8DF2 B0 03      335          BCS REC1            ;NOT NEXT LINE
8DF4 CE EB 8D   336          DEC CVT
8DF7 AD EA 8D   337  REC1    LDA CHT
8DFA 85 24      338          STA CH
8DFC AD EB 8D   339          LDA CVT
8DFF 20 5B FB   340          JSR TABV
8E02 68         341          PLA                 ;RESTORE ACC
8E03 60         342          RTS
8E04            343  *  READ OPTIONS
8E04 A0 04      344  ROPT    LDY #4              ;PRESET FLAGS
8E06 A9 00      345          LDA #0
8E08 99 8F 8E   346  RP0     STA BFG,Y
8E0B 88         347          DEY
8E0C C0 FF      348          CPY #$FF
8E0E D0 F8      349          BNE RP0
8E10 20 0C FD   350  RP05    JSR RDKEY           ;GET CHAR
8E13 C9 C2      351          CMP #$C2            ;B?
8E15 D0 01      352          BNE RP1
8E17 60         353          RTS
8E18 C9 D3      354  RP1     CMP #$D3            ;S?
8E1A D0 0F      355          BNE RP2
8E1C 18         356          CLC
8E1D AD 90 8E   357          LDA SFL
8E20 69 80      358          ADC #$80
8E22 8D 90 8E   359          STA SFL
8E25 8D 92 8E   360          STA AFL
8E28 4C 5F 8E   361          JMP RP6
8E2B C9 CC      362  RP2     CMP #$CC            ;L?
8E2D D0 13      363          BNE RP3
8E2F 18         364          CLC
8E30 AD 91 8E   365          LDA LFL
8E33 69 80      366          ADC #$80
8E35 8D 91 8E   367          STA LFL
8E38 C9 00      368          CMP #0
8E3A F0 03      369          BEQ RP25
8E3C 20 5C 92   370          JSR NUMBER          ;GET LINE RANGE
8E3F 4C 5F 8E   371  RP25    JMP RP6
8E42 C9 C1      372  RP3     CMP #$C1            ;A?
8E44 D0 0C      373          BNE RP4
8E46 18         374          CLC
8E47 AD 92 8E   375          LDA AFL
8E4A 69 80      376          ADC #$80
8E4C 8D 92 8E   377          STA AFL
8E4F 4C 5F 8E   378          JMP RP6
8E52 C9 C5      379  RP4     CMP #$C5            ;E?
8E54 D0 03      380          BNE RP5
8E56 4C 5C 93   381          JMP FINIS           ;DONE
8E59 20 3A FF   382  RP5     JSR BELL            ;ERROR
8E5C 4C 10 8E   383          JMP RP05
8E5F A9 C1      384  RP6     LDA #$C1            ;A
8E61 8D D8 06   385          STA $6D8
8E64 AD 92 8E   386          LDA AFL
8E67 F0 05      387          BEQ RP7             ;NOT SET
8E69 A9 01      388          LDA #$01            ;INV A
8E6B 8D D8 06   389          STA $6D8
```

```
0E6E A9 CC      390 RP7     LDA #$CC        ;L
0E70 8D E4 06   391         STA $6E4
0E73 AD 91 0E   392         LDA LFL
0E76 F0 05      393         BEQ RP8
0E78 A9 0C      394         LDA #$0C        ;INV C
0E7A 8D E4 06   395         STA $6E4
0E7D A9 D3      396 RP8     LDA #$D3        ;S
0E7F 8D 64 06   397         STA $664
0E82 AD 90 0E   398         LDA SFL
0E85 F0 05      399         BEQ RP9
0E87 A9 13      400         LDA #$13
0E89 8D 64 06   401         STA $664
0E8C 4C 10 0E   402 RP9     JMP RP05        ;GET NEXT CMMD
0E8F            403 BFG     DFS $1          ;FLAG STORAGE
0E90            404 SFL     DFS $1
0E91            405 LFL     DFS $1
0E92            406 AFL     DFS $1
0E93            407 EFL     DFS $1
0E94            408 *  MOVE MEMORY UP BY ONE
0E94 A0 01      409 MOVU    LDY #1
0E96 A5 B8      410         LDA TXTPTR      ;SAVE TXTPTR
0E98 48         411         PHA             ;IN THE STACK
0E99 A5 B9      412         LDA TXTPTR+1
0E9B 48         413         PHA
0E9C A5 B8      414 MU0     LDA TXTPTR
0E9E C5 AF      415         CMP PGEND       ;CHECK FOR LAST ADDRESS
0EA0 D0 17      416         BNE MU1         ;NOT LAST ADDRESS
0EA2 A5 B9      417         LDA TXTPTR+1
0EA4 C5 B0      418         CMP PGEND+1
0EA6 D0 11      419         BNE MU1         ;NOT LAST ADDRESS
0EA8 68         420         PLA
0EA9 85 B9      421         STA TXTPTR+1
0EAB 68         422         PLA             ;RESTORE TEXT POINTER
0EAC 85 B8      423         STA TXTPTR
0EAE C6 AF      424         DEC PGEND       ;DEC PGEND
0EB0 A5 AF      425         LDA PGEND
0EB2 C9 FF      426         CMP #$FF
0EB4 D0 02      427         BNE MU05
0EB6 C6 B0      428         DEC PGEND+1
0EB8 60         429 MU05    RTS
0EB9 B1 B8      430 MU1     LDA (TXTPTR),Y  ;LOAD HI BYTE
0EBB 88         431         DEY
0EBC 91 B8      432         STA (TXTPTR),Y  ;STORE BYTE
0EBE C8         433         INY
0EBF 20 00 0F   434         JSR TPINC       ;INC TEXT POINTER
0EC2 4C 9C 0E   435         JMP MU0
0EC5            436 *  MOVE MEMORY DOWN RTN
0EC5            437 *     TXTPTR RESTORED
0EC5 20 F5 0E   438 MOVD    JSR IPGD        ;INC PGEND
0EC8 A5 AF      439         LDA PGEND
0ECA 85 FD      440         STA TPTR
0ECC A5 B0      441         LDA PGEND+1
0ECE 85 FE      442         STA TPTR+1
0ED0 20 EA 0E   443         JSR DTR         ;DECREMENT TPTR
0ED3 A0 00      444 MD1     LDY #0
0ED5 B1 FD      445         LDA (TPTR),Y    ;GET BYTE
0ED7 C8         446         INY
0ED8 91 FD      447         STA (TPTR),Y    ;STORE IT
0EDA 20 EA 0E   448         JSR DTR         ;DEC TPTR
0EDD A5 FD      449         LDA TPTR
0EDF C5 B8      450         CMP TXTPTR      ;CHECK FOR LAST ADDRESS
0EE1 D0 F0      451         BNE MD1         ;NOT EQUAL
0EE3 A5 FE      452         LDA TPTR+1
0EE5 C5 B9      453         CMP TXTPTR+1
0EE7 D0 EA      454         BNE MD1         ;NOT DONE
0EE9 60         455         RTS             ;RETURN
0EEA            456 *
0EEA            457 *  DECREMENT TPTR RTN
0EEA C6 FD      458 DTR     DEC TPTR
0EEC A5 FD      459         LDA TPTR
0EEE C9 FF      460         CMP #$FF
0EF0 D0 02      461         BNE DT1
0EF2 C6 FE      462         DEC TPTR+1
0EF4 60         463 DT1     RTS
0EF5            464 *   INC PGEND
0EF5 E6 AF      465 IPGD    INC PGEND
0EF7 A5 AF      466         LDA PGEND
0EF9 C9 00      467         CMP #0
0EFB D0 02      468         BNE IP1
0EFD E6 B0      469         INC PGEND+1
0EFF 60         470 IP1     RTS
0F00            471 *  INC TXTPTR
0F00 18         472 TPINC   CLC
0F01 A5 B8      473         LDA TXTPTR
0F03 69 01      474         ADC #$1
0F05 85 B8      475         STA TXTPTR
0F07 90 02      476         BCC TP1
0F09 E6 B9      477         INC TXTPTR+1
0F0B 60         478 TP1     RTS
0F0C            479 *   PRINT TOKEN
0F0C            480 *     ON ENTRY TOKEN IS IN ACC
0F0C 8D 59 0F   481 PTOK    STA TOKT
0F0F A9 A0      482         LDA #$A0        ;PRINT BLANK
0F11 20 ED FD   483         JSR COUT
0F14 A9 B4      484         LDA #TABLE
0F16 85 FD      485         STA TPTR
0F18 A9 93      486         LDA /TABLE
0F1A 85 FE      487         STA TPTR+1
0F1C A9 80      488         LDA #$80
0F1E 8D 5A 0F   489         STA TOKT2       ;TEMP STORAGE
0F21 AD 59 0F   490 PT2     LDA TOKT        ;RELOAD TOKEN
0F24 CD 5A 0F   491         CMP TOKT2       ;TOKEN FOUND?
0F27 D0 19      492         BNE PT5
0F29 A0 00      493 PT3     LDY #0          ;NOW PRINT TOKEN USING INVFLG
0F2B B1 FD      494         LDA (TPTR),Y
0F2D C9 60      495         CMP #$60        ;BLANK?
0F2F F0 0B      496         BEQ PT4
0F31 09 80      497         ORA #$80        ;CONVERT TO NORMAL ASCII
0F33 20 ED FD   498         JSR COUT        ;NOW PRINT IT
0F36 20 C7 0D   499         JSR TINC
0F39 4C 29 0F   500         JMP PT3         ;NOW GET NEXT CHAR TILL DONE
0F3C A9 A0      501 PT4     LDA #$A0        ;PRINT A BLANK
0F3E 20 ED FD   502         JSR COUT
0F41 60         503         RTS
0F42 A0 00      504 PT5     LDY #0          ;TOKEN DOES NOT MATCH
0F44 B1 FD      505         LDA (TPTR),Y    ;NOW SEARCH FOR A BLANK
0F46 C9 60      506         CMP #$60
0F48 F0 06      507         BEQ PT7         ;BLANK FOUND
0F4A 20 C7 0D   508         JSR TINC
0F4D 4C 42 0F   509         JMP PT5
0F50 EE 5A 0F   510 PT7     INC TOKTZ
0F53 20 C7 0D   511         JSR TINC
0F56 4C 21 0F   512         JMP PT2
0F59            513 TOKT    DFS $1
0F5A            514 TOKTZ   DFS $1
0F5B            515 *
0F5B            516 *   MATCH RTN
0F5B            517 *    CARRY CLEAR - NO MATCH
0F5B            518 *
0F5B AE 75 0F   519 MATCH   LDX BOFF        ;BUFF OFFSET (0 OR $40)
0F5E 18         520         CLC             ;CLEAR CARRY ON ENTRY
0F5F A0 00      521         LDY #0
0F61 BD 87 0B   522 MA1     LDA BUFF1,X     ;GET CHAR
0F64 C9 00      523         CMP #0          ;CHK FOR END OF LINE
0F66 F0 09      524         BEQ MA2         ;DONE
0F68 D1 B8      525         CMP (TXTPTR),Y
0F6A D0 07      526         BNE MA3
0F6C C8         527         INY
0F6D E8         528         INX             ;INC OFFSETS
0F6E 4C 61 0F   529         JMP MA1         ;DO IT AGAIN
0F71 38         530 MA2     SEC             ;SET CARRY - MATCH FOUND
0F72 60         531         RTS
0F73 18         532 MA3     CLC             ;NO MATCH
0F74 60         533         RTS
0F75            534 BOFF    DFS $1          ;BUFF OFFSET
0F75            535 *  REPLACE CHECK
0F76            536 *     CARRY CLEAR - DON'T REPLACE
0F76 AD 92 0E   537 REPL    LDA AFL         ;CHECK AUTO FLAG
0F79 C9 00      538         CMP #0
0F7B F0 02      539         BEQ RE1         ;NOT AUTO
0F7D 38         540 RE0     SEC             ;REPLACE
0F7E 60         541         RTS
0F7F 20 3A FF   542 RE1     JSR BELL
0F82 A2 00      543         LDX #0
0F84 BD AB 0F   544 RE15    LDA RF,X
0F87 9D D0 06   545         STA $6D0,X
0F8A E8         546         INX
0F8B E0 04      547         CPX #$4
0F8D D0 F5      548         BNE RE15
0F8F A9 45      549         LDA #$45        ;FLASHING E
0F91 8D 70 06   550         STA $670
0F94 20 0C FD   551 RE25    JSR RDKEY       ;READ KBD
0F97 48         552         PHA             ;SAVE IT
0F98 A9 C5      553         LDA #$C5        ;NORMAL E
0F9A 8D 70 06   554         STA $670
0F9D 68         555         PLA
0F9E C9 D9      556         CMP #$D9        ;Y?
0FA0 F0 DB      557         BEQ RE0         ;REPLACE
0FA2 C9 C5      558         CMP #$C5        ;E?
0FA4 D0 03      559         BNE RE3
0FA6 4C 5C 93   560         JMP FINIS
0FA9 18         561 RE3     CLC
0FAA 60         562         RTS             ;NO REPLACE
0FAB D2 D0 CC   563 RF      ASC 'RPL?'
0FAE BF
0FAF            564 *  DIFFERENCE RTN
0FAF            565 *   RESULT IN TF AND TF+1
0FAF A9 00      566 DIFF    LDA #0
0FB1 AA         567         TAX
0FB2 A0 09      568         LDY #9          ;PRESET CNTR
0FB4 99 08 90   569 DF1     STA DIF1,Y
0FB7 88         570         DEY
0FB8 C0 FF      571         CPY #$FF        ;PRESET COUNTERS TO ZERO
0FBA D0 F8      572         BNE DF1
0FBC C8         573         INY
0FBD B9 87 0B   574 DF2     LDA BUFF1,Y     ;GET BUFF CHAR
0FC0 F0 07      575         BEQ DF3
0FC2 FE 08 90   576 DF25    INC DIF1,X      ;CHAR CNTR
0FC5 C8         577         INY
0FC6 4C BD 0F   578         JMP DF2         ;GET NEXT CHAR
0FC9 EE 0C 90   579 DF3     INC DIF5        ;BUFF CNTR
0FCC AD 0C 90   580         LDA DIF5
0FCF C9 04      581         CMP #4          ;DONE?
0FD1 F0 0E      582         BEQ DE8         ;DONE
0FD3 18         583         CLC             ;NOW PRESET BUFF OFFSET
0FD4 A9 20      584         LDA #$20        ;BUFF OFFSET
0FD6 6D 0D 90   585         ADC DIF6
0FD9 8D 0D 90   586         STA DIF6
0FDC A8         587         TAY
0FDD E8         588         INX             ;BUFF CNTR OFFSET
0FDE 4C BD 0F   589         JMP DF2         ;NOW GET NEXT BUFFER
0FE1 38         590 DE8     SEC             ;NOW CALCULATE DIFFERENCES
0FE2 AD 08 90   591         LDA DIF1
0FE5 ED 09 90   592         SBC DIF1+1      ;FIRST DIFF
0FE8 8D 0E 90   593         STA DIF7
0FEB 29 80      594         AND #$80        ;CHECK HI BIT
0FED D0 05      595         BNE DE9
0FEF A9 01      596         LDA #1
0FF1 8D 10 90   597         STA TFL         ;SET MINUS FLAG
0FF4 38         598 DE9     SEC             ;UNTOKENIZED
```

```
8FF5 AD 0A 90   599          LDA DIF1+2
8FF8 ED 0B 90   600          SBC DIF1+3      ;TOKENIZED DIFF
8FFB 8D 0F 90   601          STA DIF8
8FFE 29 80      602          AND #$80        ;CHK HI BIT
9000 D0 05      603          BNE DF10
9002 A9 01      604          LDA #1
9004 8D 11 90   605          STA TFL+1       ;TOKENIZED FLAG
9007 60         606  DF10    RTS             ;DONE
9008            607  DIF1    DFS $4          ;BUFF CNTRS
900C            608  DIF5    DFS $1          ;TEMP CNTR
900D            609  DIF6    DFS $1
900E            610  DIF7    DFS $1
900F            611  DIF8    DFS $1          ;FIND - REPL (UNTOK)
9010            612  TFL     DFS $2          ;FIND - REPL (TOK)
9012 A9 1F      613  OPTIONS LDA #OPTBL      ;1 IF DIFF >0
9014 85 FD      614          STA PNTADD      ;PRINT OPTION MENU
9016 A9 8C      615          LDA /OPTBL
9018 85 FE      616          STA PNTADD+1
901A A9 0A      617          LDA #10
901C 85 24      618          STA CH          ;MOVE CUR OVER
901E 20 78 FB   619          JSR PRINT
9021 A9 05      620          LDA #$05        ;MOVE CUR
9023 85 24      621          STA CH
9025 20 04 0E   622          JSR ROPT
9028 20 AF 8F   623          JSR DIFF        ;CALCULATE DIFFERENCES
902B A9 00      624          LDA #0          ;SET WINDOW BOTTOM
902D 8D 5B 92   625          STA QFLG        ;SET QUOTE FLAG
9030 8D 41 92   626          STA RFLG        ;SET REM FLAG
9033 85 24      627          STA CH
9035 A9 0F      628          LDA #15
9037 20 5B FB   629          JSR TABV
903A A9 10      630          LDA #16
903C 8D 23 00   631          STA WNDBTM
903F A0 00      632          LDY #0
9041 B9 67 00   633          LDA $67,Y       ;START OF PROGRAM
9044 85 B8      634          STA TXTPTR
9046 B9 68 00   635          LDA $68,Y       ;PUT IN TXTPTR
9049 85 B9      636          STA TXTPTR+1
904B A5 B8      637  OP1     LDA TXTPTR      ;SAVE TEXT POINTER IN TPTR
904D 8D 06 92   638          STA NXLN
9050 A5 B9      639          LDA TXTPTR+1
9052 8D 07 92   640          STA NXLN+1      ;SAVE NEXT ADDRESS
9055 A0 01      641          LDY #1          ;CHECK FOR LAST LINE
9057 B1 B8      642          LDA (TXTPTR),Y  ;HIGH BYTE
9059 D0 03      643          BNE OP2         ;NOT LAST LINE
905B 4C 5C 93   644          JMP FINIS       ;DONE
905E 20 16 93   645  OP2     JSR LNCHK       ;CHECK FOR LFL
9061 90 1A      646          BCC OP25        ;WITHIN LINE RANGE
9063 A0 00      647          LDY #0          ;SET TXTPTR TO NEXT LINE
9065 B1 B8      648          LDA (TXTPTR),Y
9067 8D 08 92   649          STA NP          ;TEMP STORAGE
906A C8         650          INY
906B B1 B8      651          LDA (TXTPTR),Y
906D 8D 09 92   652          STA NP+1
9070 AD 08 92   653          LDA NP
9073 85 B8      654          STA TXTPTR
9075 AD 09 92   655          LDA NP+1
9078 85 B9      656          STA TXTPTR+1
907A 4C 4B 90   657          JMP OP1         ;NEXT LINE
907D AD 90 0E   658  OP25    LDA SFL         ;CHECK SUP FL
9080 D0 10      659          BNE OP27
9082 A0 02      660          LDY #2          ;NOW PRINT LINE NUMBER
9084 B1 B8      661          LDA (TXTPTR),Y
9086 AA         662          TAX
9087 C8         663          INY
9088 B1 B8      664          LDA (TXTPTR),Y
908A 20 24 ED   665          JSR $ED24       ;PRINT LINE NUMBER
908D A9 A0      666          LDA #$A0        ;PRINT BLANK
908F 20 ED FD   667          JSR COUT
9092 A0 04      668  OP27    LDY #4          ;NOW SET TXTPTR TO FIRST CHAR
9094 20 00 8F   669  OP3     JSR TPINC       ;INC TXTPTR BY FOUR
9097 88         670          DEY
9098 D0 FA      671          BNE OP3
909A A9 40      672          LDA #$40        ;PRESET OFFSET
909C 8D 75 8F   673          STA BOFF
909F A0 00      674  OP4     LDY #0          ;CHK FOR REM AND *
90A1 B1 B8      675          LDA (TXTPTR),Y
90A3 C9 00      676          CMP #0          ;CHECK FOR END OF LINE
90A5 D0 41      677          BNE OP48        ;NOT END OF LINE
90A7 20 8E FD   678          JSR CROUT
90AA 20 00 8F   679          JSR TPINC
90AD A9 40      680          LDA #$40        ;RESET OFFSETS
90AF 8D 75 8F   681          STA BOFF
90B2 A9 00      682          LDA #0
90B4 8D 5B 92   683          STA QFLG
90B7 8D 41 92   684          STA RFLG
90BA 4C 4B 90   685          JMP OP1         ;GET NEXT LINE
90BD C9 B2      686  OP42    CMP #$B2        ;REM?
90BF D0 03      687          BNE OP43
90C1 20 36 92   688          JSR REM         ;SET REM
90C4 C9 83      689  OP43    CMP #$83        ;DATA?
90C6 D0 03      690          BNE OP45
90C8 20 3B 92   691          JSR DATA
90CB C9 22      692  OP45    CMP #$22        ;" MARK?
90CD D0 03      693          BNE OP47
90CF 20 42 92   694          JSR QUMARK
90D2 C9 3A      695  OP47    CMP #$3A        ;COLON?
90D4 D0 42      696          BNE OP66        ;NO
90D6 AD 5B 92   697          LDA QFLG        ;CHK FOR WITHIN " "
90D9 D0 0A      698          BNE OP475
90DB AD 41 92   699          LDA RFLG        ;CHK REM FLAG
90DE D0 05      700          BNE OP475
90E0 A9 40      701          LDA #$40
90E2 8D 75 8F   702          STA BOFF        ;RESET BUFF OFFSET
90E5 4C 18 91   703  OP475   JMP OP66

90E8 20 5B 8F   704  OP48    JSR MATCH
90EB 90 0B      705          BCC OP5         ;NO MATCH
90ED 20 25 91   706          JSR PMATCH      ;PRINT MATCH IN INVERSE
90F0 20 76 8F   707          JSR REPL        ;REPLACE?
90F3 90 03      708          BCC OP5         ;DONT REPLACE
90F5 20 46 91   709          JSR FANDR       ;GO REPLACE IT
90F8 A0 00      710  OP5     LDY #0
90FA AD 90 0E   711          LDA SFL         ;CHK SUPP FLAG
90FD C9 00      712          CMP #0
90FF F0 03      713          BEQ OP55        ;NOT SET
9101 4C 11 91   714          JMP OP65        ;SUPP PRINT
9104 B1 B8      715  OP55    LDA (TXTPTR),Y  ;GET CHAR
9106 48         716          PHA             ;SAVE IT
9107 29 80      717  OP6     AND #$80        ;CHK FOR TOKEN
9109 D0 13      718          BNE OP7         ;NOT TOKEN
910B 68         719          PLA             ;GET CHAR
910C 09 80      720          ORA #$80        ;CONVERT ASCII
910E 20 ED FD   721          JSR COUT
9111 A0 00      722  OP65    LDY #0          ;GET CHAR BACK
9113 B1 B8      723          LDA (TXTPTR),Y
9115 4C BD 90   724          JMP OP42        ;CHK FOR REM DATA AND "
9118 20 00 8F   725  OP66    JSR TPINC       ;INC TXTPTR
911B 4C 9F 90   726          JMP OP4
911E 68         727  OP7     PLA             ;GET CHAR
911F 20 0C 8F   728          JSR PTOK        ;PRINT TOKEN
9122 4C 11 91   729          JMP OP65
9125            730  *  PRINT MATCH IN INVERSE
9125 AD 90 0E   731  PMATCH  LDA SFL         ;PRINT MATCH
9128 C9 00      732          CMP #0          ;FIRST CHK SUPP FL
912A F0 01      733          BEQ PM0         ;NOT SET
912C 60         734          RTS
912D 20 DD 8D   735  PM0     JSR SVCUR       ;SAVE CUR POS
9130 A0 00      736          LDY #0
9132 B9 87 8B   737  PM1     LDA BUFF1,Y
9135 C9 00      738          CMP #0          ;END?
9137 F0 09      739          BEQ PM2
9139 29 3F      740          AND #$3F        ;CONV TO INVERSE
913B 20 ED FD   741          JSR COUT
913E C8         742          INY
913F 4C 32 91   743          JMP PM1         ;GET NEXT CHAR
9142 20 EC 8D   744  PM2     JSR RECUR       ;RESTORE CURSOR
9145 60         745          RTS
9146            746  *   FOUND AND REPLACE RTN
9146            747  *    SHIFTS MEMORY AND INSERTS STRING
9146 A0 00      748  FANDR   LDY #0
9148 AD 75 8F   749          LDA BOFF        ;GET BUFF OFFSET
914B C9 00      750          CMP #0
914D F0 01      751          BEQ FA1
914F C8         752          INY             ;GET RIGHT MINUS FLAG
9150 B9 10 90   753  FA1     LDA TFL,Y
9153 8D 60 91   754          STA SOFF        ;TEMP STORAGE
9156 B9 0E 90   755          LDA DIF7,Y      ;GET RIGHT DIFF
9159 8D 61 91   756          STA TDIF        ;TEMP STORAGE
915C 20 62 91   757          JSR MEMSHT      ;MEMORY SHIFT
915F 60         758          RTS             ;DONE
9160            759  SOFF    DFS $1          ;TEMP MINUS STORAGE
9161            760  TDIF    DFS $1          ;TEMP DIFF STORAGE
9162            761  *  MEMORY SHIFT
9162            762  *   CORRECTS POINTERS, SHIFTS MEMORY
9162            763  *    AND REPLACES STRING
9162 20 EF 91   764  MEMSHT  JSR TPSAV        ;SAVE TXTPTR AND TPTR IN STACK
9165 A0 00      765  MEO     LDY #0
9167 B1 FD      766          LDA (TPTR),Y    ;STORE NEXT ADDR IN TXTPTR
9169 85 B8      767          STA TXTPTR
916B C8         768          INY
916C B1 FD      769          LDA (TPTR),Y
916E 85 B9      770          STA TXTPTR+1
9170 C9 00      771          CMP #0
9172 D0 0A      772          BNE ME1         ;LAST ADDRESS CHECK
9174 20 0A 92   773          JSR TPRES       ;RESTORE TXTPTR AND TPTR
9177 20 CD 91   774          JSR MSHIFT      ;SHIFT MEM UP OR DOWN
917A 20 15 92   775          JSR STRREP      ;REPLACE STRING
917D 60         776          RTS
917E A0 00      777  ME1     LDY #0
9180 AD 60 91   778          LDA SOFF        ;0 FOR MEM DOWN
9183 C9 00      779          CMP #$0         ;ZERO?
9185 D0 06      780          BNE ME13        ;SUBTRACT
9187 20 9B 91   781          JSR ADD
918A 4C 90 91   782          JMP ME16
918D 20 BA 91   783  ME13    JSR SUB
9190 A5 B8      784  ME16    LDA TXTPTR      ;STORE TXTPTR IN TPTR
9192 85 FD      785          STA TPTR
9194 A5 B9      786          LDA TXTPTR+1
9196 85 FE      787          STA TPTR+1
9198 4C 65 91   788          JMP MEO
919B            789  *  ADD RTN - Y=0 ON ENTRY
919B            790  *   MOVES POINTERS DOWN BY TWO COMPLEMENT OF TDIF
919B AD 61 91   791  ADD     LDA TDIF        ;GET TDIF
919E 49 FF      792          EOR #$FF        ;CONVERT TO POS NUM
91A0 18         793          CLC
91A1 69 01      794          ADC #$1
91A3 8D B9 91   795          STA TDTEMP      ;TEMP STORAGE
91A6 18         796          CLC
91A7 B1 FD      797          LDA (TPTR),Y    ;GET LO BYTE
91A9 6D B9 91   798          ADC TDTEMP      ;ADD OFFSET
91AC 91 FD      799          STA (TPTR),Y    ;NOW STORE IT
91AE 90 08      800          BCC ADJ         ;INC HI BYTE?
91B0 C8         801          INY
91B1 B1 FD      802          LDA (TPTR),Y    ;YES - GET HI BYTE
91B3 18         803          CLC
91B4 69 01      804          ADC #$1         ;ADD 1
91B6 91 FD      805          STA (TPTR),Y
91B8 60         806  ADJ     RTS             ;DONE
91B9            807  TDTEMP  DFS $1          ;CONVERTED TDIF
91BA            808  *  SUB RTN   Y REG = 0 ON ENTRY
```

```
91BA            809 *   MOVES POINTERS  UP  BY TDIF
91BA 38         810 SUB     SEC
91BB B1 FD      811         LDA (TPTR),Y        ;GET LO BYTE
91BD ED 61 91   812         SBC TDIF            ;SUB OFFSET
91C0 91 FD      813         STA (TPTR),Y        ;SAVE IT
91C2 B0 08      814         BCS SU3
91C4 C8         815         INY
91C5 B1 FD      816         LDA (TPTR),Y        ;INC HI BYTE
91C7 18         817         CLC
91C8 69 FF      818         ADC #$FF            ;SUB 1
91CA 91 FD      819         STA (TPTR),Y        ;SAVE IT
91CC 60         820 SU3     RTS                 ;DONE
91CD            821 *  MSHIFT RTN  SHIFT MEM UP OR DOWN
91CD AD 60 91   822 MSHIFT  LDA SOFF            ;MINUS FLAG
91D0 F0 0F      823         BEQ MS6
91D2 AD 61 91   824 MS1     LDA TDIF            ;GET DIFF
91D5 F0 09      825         BEQ MS2
91D7 20 94 8E   826         JSR MOVU
91DA CE 61 91   827         DEC TDIF            ;INC TDIF
91DD 4C D2 91   828         JMP MS1
91E0 60         829 MS2     RTS
91E1 AD B9 91   830 MS6     LDA TDTEMP
91E4 F0 FA      831         BEQ MS2             ;DONE
91E6 20 C5 8E   832         JSR MOVD
91E9 CE B9 91   833         DEC TDTEMP
91EC 4C E1 91   834         JMP MS6             ;DO IT AGAIN
91EF            835 *  TPSAV RTN - SAVE TXTPTR AND TPINC IN STACK
91EF AD 06 92   836 TPSAV   LDA NXLN            ;SET TPTR
91F2 85 FD      837         STA TPTR
91F4 AD 07 92   838         LDA NXLN+1
91F7 85 FE      839         STA TPTR+1
91F9 A5 B8      840         LDA TXTPTR
91FB 8D 04 92   841         STA TPTEMP
91FE A5 B9      842         LDA TXTPTR+1
9200 8D 05 92   843         STA TPTEMP+1
9203 60         844         RTS
9204            845 TPTEMP  DFS $2              ;TXTPTR
9206            846 NXLN    DFS $2              ;NEXT ADDRESS PNTR
9208            847 NP      DFS $2              ;TEMP STORAGE FOR NEXT ADDRESS
920A            848 *  TPRES RTN - RESTORE TPTR AND TXTPTR
920A AD 05 92   849 TPRES   LDA TPTEMP+1
920D 85 B9      850         STA TXTPTR+1
920F AD 04 92   851         LDA TPTEMP
9212 85 B8      852         STA TXTPTR
9214 60         853         RTS
9215            854 *  STRING REPLACE RTN
9215 20 EF 91   855 STRREP  JSR TPSAV           ;SAVE POINTERS
9218 20 9C FC   856         JSR CLEOL           ;CLEAR LINE
921B A0 00      857         LDY #0
921D AD 75 BF   858         LDA BOFF            ;GET OFFSET
9220 18         859         CLC
9221 69 20      860         ADC #$20            ;ADD 20 TO GET TO REPL BUFF
9223 AA         861         TAX
9224 BD 87 BB   862 ST1     LDA BUFF1,X         ;GET REPLACE CHAR
9227 C9 00      863         CMP #0              ;DONE?
9229 D0 04      864         BNE ST22
922B 20 0A 92   865         JSR TPRES
922E 60         866         RTS
922F 91 B8      867 ST22    STA (TXTPTR),Y      ;STORE CHAR
9231 E8         868         INX                 ;BUFF INC
9232 C8         869         INY
9233 4C 24 92   870         JMP ST1
9236            871 *   REM RTN
9236 A9 01      872 REM     LDA #1
9238 8D 41 92   873         STA RFLG            ;SET REM FLAG
923B A9 00      874 DATA    LDA #0              ;SET BUFFER OFFSET
923D 8D 75 BF   875         STA BOFF
9240 60         876         RTS
9241            877 RFLG    DFS $1              ;REM FLAG
9242            878 *  QUOTE MARK RTN
9242 AD 5B 92   879 QUMARK  LDA QFLG            ;SEE IF FLAG IS SET
9245 F0 0B      880         BEQ QM5             ;NOT SET
9247 A9 00      881         LDA #0              ;RESET FLAG
9249 8D 5B 92   882         STA QFLG
924C A9 40      883         LDA #$40            ;RESET BOFF
924E 8D 75 BF   884         STA BOFF
9251 60         885         RTS
9252 EE 5B 92   886 QM5     INC QFLG            ;SET QUOTE FLAG
9255 A9 00      887         LDA #0              ;SET BOFF
9257 8D 75 BF   888         STA BOFF
925A 60         889         RTS                 ;DONE
925B            890 QFLG    DFS $1              ;QUOTE FLAG
925C            891 *  NUMBER - INPUT LINE RANGE
925C 20 DD 8D   892 NUMBER  JSR SVCUR           ;SAVE CUR POS
925F A9 18      893         LDA #$18
9261 8D 23 00   894         STA WNDBTM
9264 A9 00      895         LDA #0
9266 8D 11 93   896         STA NOFF
9269 85 24      897         STA CH
926B A2 00      898         LDX #0              ;PRINT BEG#
926D BD C8 92   899 NU1     LDA BF,X
9270 9D D0 06   900         STA $6D0,X
9273 E8         901         INX
9274 E0 04      902         CPX #4
9276 D0 F5      903         BNE NU1
9278 A9 00      904         LDA #0
927A 85 24      905         STA CH
927C 18         906         CLC
927D A5 25      907         LDA CV
927F 69 01      908         ADC #1
9281 20 5B FB   909         JSR TABV
9284 20 D0 92   910         JSR GETNUM          ;GET LINE NUM
9287 A9 00      911         LDA #0
9289 85 24      912         STA CH              ;RESET CURSOR
928B A2 00      913         LDX #0              ;PRINT END#
928D BD C4 92   914 NU2     LDA EF,X
9290 9D D0 06   915         STA $6D0,X
9293 E8         916         INX
9294 E0 04      917         CPX #4
9296 D0 F5      918         BNE NU2
9298 EE 11 93   919         INC NOFF            ;INC OFFSET
929B EE 11 93   920         INC NOFF
929E 20 9C FC   921         JSR CLEOL
92A1 20 D0 92   922         JSR GETNUM
92A4 A9 00      923         LDA #0
92A6 85 24      924         STA CH
92A8 20 9C FC   925         JSR CLEOL           ;CLEAR NUM
92AB A2 00      926         LDX #0
92AD BD CC 92   927 NU3     LDA CF,X
92B0 9D D0 06   928         STA $6D0,X
92B3 E8         929         INX
92B4 E0 04      930         CPX #4
92B6 D0 F5      931         BNE NU3
92B8 EE EB 8D   932         INC CVT
92BB 20 EC 8D   933         JSR RECUR           ;RESTORE CURSOR
92BE A9 10      934         LDA #16             ;RESET WINDOW BTM
92C0 8D 23 00   935         STA WNDBTM
92C3 60         936         RTS                 ;DONE
92C4 C5 CE C4   937 EF      ASC "END#"
92C7 A3
92C8 C2 C5 C7   938 BF      ASC "BEG#"
92CB A3
92CC C3 CD CD   939 CF      ASC "CMMD"
92CF C4
92D0            940 * GETNUM - INPUT LINE #
92D0 20 EF 91   941 GETNUM  JSR TPSAV           ;SAVE TXTPTR
92D3 A0 00      942 GE1     LDY #0
92D5 20 0C FD   943 GE3     JSR RDKEY           ;GET CHAR
92D8 C9 8D      944         CMP #$8D            ;CR?
92DA F0 19      945         BEQ GE6
92DC C9 88      946         CMP #$88            ;BA?
92DE D0 0A      947         BNE GE5
92E0 88         948         DEY
92E1 C0 FF      949         CPY #$FF            ;BEFORE START?
92E3 F0 EE      950         BEQ GE1             ;YES
92E5 C6 24      951         DEC CH              ;MOVE CURSOR
92E7 4C D5 92   952         JMP GE3             ;GET NEXT CHAR
92EA 20 ED FD   953 GE5     JSR COUT            ;PRINT CHAR
92ED 29 7F      954         AND #$7F            ;CONVERT ASCII
92EF 91 B8      955         STA (TXTPTR),Y
92F1 C8         956         INY
92F2 4C D5 92   957         JMP GE3             ;GET NEXT CHAR
92F5 A9 00      958 GE6     LDA #0              ;DELIMIT LINE
92F7 91 B8      959         STA (TXTPTR),Y
92F9 20 7B DD   960         JSR FRMEVL          ;CONVERT TO INTEGER
92FC 20 52 E7   961         JSR GETADR
92FF AE 11 93   962         LDX NOFF            ;GET OFFSET
9302 A5 50      963         LDA LINUM
9304 9D 12 93   964         STA LN,X
9307 E8         965         INX
9308 A5 51      966         LDA LINUM+1
930A 9D 12 93   967         STA LN,X            ;STORE LINE NUMBER
930D 20 0A 92   968         JSR TPRES           ;RESTORE TXTPTR
9310 60         969         RTS
9311            970 NOFF    DFS $1
9312            971 LN      DFS $4              ;STORAGE FOR LINE NUM
9316            972 *  LINE RANGE CHECK RTN
9316            973 *   CLC - WITHIN RANGE
9316 AD 91 8E   974 LNCHK   LDA LFL
9319 D0 02      975         BNE LNC0            ;LFL SET
931B 18         976         CLC
931C 60         977         RTS
931D 20 27 93   978 LNC0    JSR LOLN            ;CHK LOW LINE #
9320 90 01      979         BCC LNC1
9322 60         980         RTS                 ;CARRY SET
9323 20 40 93   981 LNC1    JSR HILN            ;CHK HI LINE
9326 60         982         RTS                 ;CARRY SET ACCORDINGLY
9327 A0 03      983 LOLN    LDY #$3
9329 B1 B8      984         LDA (TXTPTR),Y      ;HI BYTE
932B CD 13 93   985         CMP LN+1
932E F0 04      986         BEQ LO4             ;LOW LINE BYTE SAME
9330 90 0C      987         BCC LO5             ;NOT PAST LOW NUM
9332 B0 08      988         BCS LO6             ;PAST LOW NUM
9334 88         989 LO4     DEY                 ;HI BYTE SAME
9335 B1 B8      990         LDA (TXTPTR),Y      ;CHK LO BYTE
9337 CD 12 93   991         CMP LN
933A 90 02      992         BCC LO5
933C 18         993 LO6     CLC
933D 60         994         RTS
933E 38         995 LO5     SEC
933F 60         996         RTS
9340 A0 03      997 HILN    LDY #$3             ;HI LINE
9342 B1 B8      998         LDA (TXTPTR),Y
9344 CD 15 93   999         CMP LN+3
9347 F0 05      1000        BEQ HI4             ;HI BYTE EQUAL
9349 B0 0D      1001        BCS HI6             ;PAST RANGE
934B 4C 5A 93   1002        JMP HI5             ;WITHIN RANGE
934E 88         1003 HI4    DEY
934F B1 B8      1004        LDA (TXTPTR),Y      ;LO BYTE
9351 CD 14 93   1005        CMP LN+2
9354 F0 04      1006        BEQ HI5
9356 90 02      1007        BCC HI5
9358 38         1008 HI6    SEC
9359 60         1009        RTS                 ;NOT IN RANGE
935A 18         1010 HI5    CLC
935B 60         1011        RTS                 ;IN RANGE
935C            1012 *  EXIT RTN
935C A9 18      1013 FINIS  LDA #$18            ;RESET WIN BTM
```

APPLESOFT 全域編輯器

```
935E 8D 23 00  1014          STA WNDBTM
9361 A5 AF     1015          LDA PGEND
9363 85 6D     1016          STA STREND          ;SET STREND
9365 85 69     1017          STA VARTAB          ;RESET VARTAB
9367 85 6B     1018          STA ARYTAB          ;RESET ARYTAB
9369 A5 B0     1019          LDA PGEND+1
936B 85 6E     1020          STA STREND+1
936D 85 6A     1021          STA VARTAB+1
936F 85 6C     1022          STA ARYTAB+1
9371 A9 12     1023          LDA #18             ;CLEAR BOTTOM OF SCREEN
9373 20 5B FB  1024          JSR TABV
9376 A9 10     1025          LDA #16
9378 85 22     1026          STA WNDTOP
937A 20 58 FC  1027          JSR HOME
937D A9 00     1028          LDA #0
937F 85 22     1029          STA WNDTOP
9381 4C D0 03  1030          JMP $3D0            ;RET TO BASIC
9384 45 4E 44  1031  TABLE   BLK "END FOR NEXT DATA INPUT DEL DIM READ GR TEXT
*
9387 60 46 4F
938A 52 60 4E
938D 45 58 54
9390 60 44 41
9393 54 41 60
9396 49 4E 50
9399 55 54 60
939C 44 45 4C
939F 60 44 49
93A2 4D 60 52
93A5 45 41 44
93A8 60 47 52
93AB 60 54 45
93AE 58 54 60
93B1 50 52 23  1032          HEX 50522360494E2360
93B4 60 49 4E
93B7 23 60
93B9 43 41 4C  1033          BLK "CALL PLOT HLIN VLIN HGR2 HGR "
93BC 4C 60 50
93BF 4C 4F 54
93C2 60 48 4C
93C5 49 4E 60
93C8 56 4C 49
93CB 4E 60 48
93CE 47 52 72
93D1 60 48 47
93D4 52 60
93D6 48 43 4F  1034          BLK "HCOLOR"
93D9 4C 4F 52
93DC 3D 60     1035          HEX 3D60
93DE 48 50 4C  1036          BLK "HPLOT "
93E1 4F 54 60
93E4 44 52 41  1037          BLK "DRAW XDRAW HTAB HOME ROT"
93E7 57 60 58
93EA 44 52 41
93ED 57 60 48
93F0 54 41 42
93F3 60 48 4F
93F6 4D 45 60
93F9 52 4F 54
93FC 3D 60     1038          HEX 3D60
93FE 53 43 41  1039          BLK "SCALE"
9401 4C 45
9403 3D 60     1040          HEX 3D60
9405 53 48 4C  1041          BLK "SHLOAD TRACE NOTRACE "
9408 4F 41 44
940B 60 54 52
940E 41 43 45
9411 60 4E 4F
9414 54 52 41
9417 43 45 60
941A 4E 4F 52  1042          BLK "NORMAL INVERSE FLASH COLOR"
941D 4D 41 4C
9420 60 49 4E
9423 56 45 52
9426 53 45 60
9429 46 4C 41
942C 53 48 60
942F 43 4F 4C
9432 4F 52
9434 3D        1043          HEX 3D
9435 60 50 4F  1044          BLK " POP VTAB HIMEM"
9438 50 60 56
943B 54 41 42
943E 60 48 49
9441 4D 45 4E
9444 3A 60     1045          HEX 3A60
9446 4C 4F 4D  1046          BLK "LOMEM"
9449 45 4E
944B 3A 60     1047          HEX 3A60
944D 4F 4E 45  1048          BLK "ONERR RESUME RECALL STORE SPEED"
9450 52 52 60
9453 52 45 53
9456 53 4D 45
9459 60 52 45
945C 43 41 4C
945F 4C 60 53
9462 54 4F 52
9465 45 60 53
9468 50 45 45
946B 44
946C 3D 60     1049          HEX 3D60
946E 4C 45 54  1050          BLK "LET GOTO RUN IF "
9471 60 47 4F
9474 54 4F 60
9477 52 55 4E
947A 60 49 46
947D 60
947E 52 45 53  1051          BLK "RESTORE "
9481 54 4F 52
9484 45 60
9486 26 60     1052          HEX 2660
9488 47 4F 53  1053          BLK "GOSUB RETURN REM STOP ON WAIT LOAD SAVE "
948B 55 42 60
948E 52 45 54
9491 55 52 4E
9494 60 52 45
9497 4D 60 53
949A 54 4F 50
949D 60 4F 4E
94A0 60 57 41
94A3 49 54 60
94A6 4C 4F 41
94A9 44 60 53
94AC 41 56 45
94AF 60
94B0 44 45 46  1054          BLK "DEF POKE PRINT CONT LIST CLEAR GET NEW TAB"
94B3 60 50 4F
94B6 4B 45 60
94B9 50 52 49
94BC 4E 54 60
94BF 43 4F 4E
94C2 54 60 4C
94C5 49 53 54
94C8 60 43 4C
94CB 45 41 52
94CE 60 47 45
94D1 54 60 4E
94D4 45 57 60
94D7 54 41 42
94DA 28 60     1055          HEX 2860
94DC 54 4F 60  1056          BLK "TO FN "
94DF 46 4E 60
94E2 53 50 43  1057          BLK "SPC"
94E5 28 60     1058          HEX 2860
94E7 54 48 45  1059          BLK "THEN AT NOT STEP "
94EA 4E 60 41
94ED 54 60 4E
94F0 4F 54 60
94F3 50 60 54 45
94F6 50 60
94F8 2B 60 2D  1060
94FB 60 2A 60
94FE 2F 60 5E
9501 60
9502 41 4E 44  1061          BLK "AND OR "
9505 60 4F 52
9508 60
9509 3E 60 3D  1062          HEX 3E603D603C60
950C 60 3C 60
950F 53 47 4E  1063          BLK "SGN INT "
9512 60 49 4E
9515 54 60
9517 41 42 53  1064          BLK "ABS USR FRE SCRN"
951A 60 55 53
951D 52 60 46
9520 52 45 60
9523 53 43 52
9526 4E
9527 28 60     1065          HEX 2860
9529 50 44 4C  1066          BLK "PDL POS SQR RND LOG EXP COS SIN TAN "
952C 60 50 4F
952F 53 60 53
9532 51 52 60
9535 52 4E 44
9538 60 4C 4F
953B 47 60 45
953E 58 50 60
9541 43 4F 53
9544 60 53 49
9547 4E 60 54
954A 41 4E 60
954D 41 54 4E  1067          BLK "ATN PEEK LEN STR"
9550 60 50 45
9553 45 4B 60
9556 4C 45 4E
9559 60 53 54
955C 52
955D 24 60     1068          HEX 2460
955F 56 41 4C  1069          BLK "VAL ASC CHR"
9562 60 41 53
9565 43 60 43
9568 48 52
956A 24 60     1070          HEX 2460
956C 4C 45 46  1071          BLK "LEFT"
956F 54
9570 24 60     1072          HEX 2460
9572 52 49 47  1073          BLK "RIGHT"
9575 48 54
9577 24 60     1074          HEX 2460
9579 4D 49 44  1075          BLK "MID"
957C 24 60     1076          HEX 2460
957E           1077          END

##### END OF ASSEMBLY
```

機械語言程式輸入方法

一些初入門的讀者却不太明白應該怎樣輸入機械碼程式其實輸入的方法十分簡單；但輸入機械碼程式之前，我們必須先進入電腦的監督系統（MONITOR）。假若讀者擁有的電腦是APⅡ，則只須從鍵盤按入 CALL －151 指令就可從 APPLESOFT 進入監督系統；這時游標（CURSOR）的左方出現一個星號＊，這個星號就是監督系統的標示字，代表電腦已進入監督系統。這之後，我們就可以將正確的字組（BYTE）放進適當的位置裡。先按入地址，然後是冒號：；跟着便按入所需字組內容，再按下〔RETURN〕便可。例如我們想將6D這個字組放進9088這個地址內只需按入

＊9088：6D〔RETURN〕

假如需要將一系列的字組內容放進連續的地址內，我們無須將每個地址逐一按入，只須按入起始地址及冒號，然後將每個字組逐一按入，核對無誤後便可按下〔RETURN〕，這些字組便會自動被貯入連續的地址內。例如我們想將6D、00、D4、00…… 等字組分別放進 9088、9089、908A、908B……等地址內，只須按入

＊9088：6D 00 D4 00……

當一切字組正確輸入後，便可按下〔RETURN〕。有兩點讀者必須要留意的：第一，輸入的字組與字組之間必須以一空位分隔開，不能連在一起。第二，每次輸入的字組不能太多。當我們在電腦螢幕上輸入第七行字組時（大約是第82個字組），電腦會發出嘟、嘟聲，表示不能再繼續輸入，否則電腦會自動取消剛才的所有輸入字組。這時我們便要按下〔RETURN〕，所有字組便會被貯進各地址內。假如仍然有其他字組需要輸入，按照這方法繼續輸入便可。若要電腦顯示出你剛才所輸入的字組內容，只須按入起始地址然後按下〔RETURN〕。每按一次〔RETURN〕，電腦會將八個連續地址的字組內容顯示出來。假若我們想知道兩個地址之間的字組內容，亦可按入＜起始地址＞·＜結束地址＞然後按〔RETURN〕，電腦便會一次過將這兩個地址之間的字組內容全部顯示出來。例如我們想知道地址 9088 至94A0之間的內容，只須按入

＊9088·94A0 然後按〔RETURN〕

當完成一切輸入後，我們當然希望將這些資料儲存起來，以便日後使用。讀者可選擇儲存在磁碟或磁帶上。

假若讀者使用是磁碟，儲存的方法是按入 BSAVE ＜存檔名稱＞，A$＜起始地址＞，L$＜字組長度＞然後再按下〔RETURN〕。例如我們要將地址 9088 至94A0 之間的字組內容用 SHAPE 這個存檔名稱來儲存在磁碟上，由於這裡共有 419（94A0減 9088＋六進制）個字組，所以要按照以下的方法輸入：

BSAVE SHAPE，A$ 9088，L$419 然後按〔RETURN〕

$ 符號以後的數字是十六進制的。假若讀者使用十進位數字，可刪去$符號。例如：

BSAVE SHAPE，A$ 9088，L1045 跟上面的指令句是完全一樣的。還有一點要提醒讀者的，就是在打入這句指令前必須先跳出監督系統，因爲在監督系統內是不會接納這樣的指令句的。跳出監督系統的方法是按〔RESET〕（有些 MODEL 要同時按〔CTRL〕及〔RESET〕）。

要讀回磁碟上的機械碼資料，只須按入 BLOAD ＜存檔名稱＞，然後按〔RETURN〕。例如：

BLOAD SHAPE 然後按〔RETURN〕這樣，電腦就會自動將存檔 SHAPE內的資料放在它們原來的地址。假若我們要將它放在其它地址內，可按入 BLOAD ＜存檔名稱＞，A$＜起始地址＞，然後再按〔RETURN〕。例如：

BLOAD SHAPE，A$8000

就是將 SHAPE 內的資料放在8000（十六進位）以後的地址內。讀者若使用十進位，可刪去$符號。■

備有 CHECK SUM 絕無打錯之虞

大家在輸入機械碼時可能會有打錯數字的可能，爲方便偵錯，我們出版一個CHECK SUM 程式專門用來幫助大家核對電腦時代每期所刊載的機械碼程式。CHECK SUM 的用法是假定每行地址有8組數值，例如：

9088：1C 2B 3A 4D 5E 6F 7A 8B

當 CHECK SUM 檢視這行地址上的數值後，會產生一個參數值，例如60。我們會將電腦時代上所列載的機械碼程式全部用CHECK SUM來執行一次，每個都產生一個CHECK 錯參數。大家於打入機械碼列表時，可用來參考比較，假如輸入數值完全正確，則所產生出來的參數值必定是相同，相反，假如任何其中一個數值輸入錯誤，則參數值一定不同，如此一來，便可很快看到究竟程式有沒有打錯。

這個CHECK SUM 軟件訂價60元港幣，其中包括十二期免費CHECK TABLE 通訊。大家可到電腦時代讀者服務部選擇。

每個CHECK SUM 軟件將會隨同附送一個APPLESOFT 程式語法偵錯器，方便大家用來檢視APPLESOFT BASIC程式的偵錯。■

訂閱辦法

訂閱COMPUSOFT 磁碟月刊辦法簡單，訂戶每期收到磁碟後，自行COPY一份，然後寄回磁碟給我們，我們便會在下期出版之時繼續寄出。當然，訂戶亦可於每月出版之時前來電腦時代出版社讀者服務部攞取每期月刊。

讀者如對訂閱辦法有任何疑問，可致電3-312007 查詢。

海外讀者只接受連同磁碟的訂閱，請附加空郵費用20%。澳門讀者亦只接受連同磁碟的訂閱，請附加多10%郵費。

只適用於本港訂戶

本人欲訂閱 COMPUSOFT 磁碟月刊

- ☐ 12期　HK $180（退回磁碟）
- ☐ 12期　HK $540（連同磁碟）*
- ☐ 6期　HK $100（退回磁碟）
- ☐ 6期　HK $280（連同磁碟）*
- ☐ 1期試閱　HK $ 50（連同磁碟）*

由第____期開始

支票收款人請填寫 "Computing Age Publisher"

請寄九龍中央郵箱71193號

信封面請註明「訂閱」磁碟月刊

＊磁碟歸訂戶擁有

姓名：________________

地址：________________

電話：________________

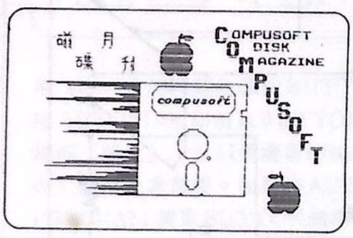
附上劃線支票 No. __________

（請用英文填寫，方便電腦處理）

NO.__________

Official Use

第3期



大逃亡（動畫卡通大賞）

看過大逃亡這個卡通動畫，你會驚覺蘋果電腦在模擬動畫方面的能力，大家可以細心研究這個程式的編寫技巧。

陽曆查陰曆日子修訂版

這個修訂版本，加添了生肖及時辰的顯示，進一步加強本程式的能力。日後將會有一系列的程式應用這個程式，各位萬勿錯過。

你是否樂觀的人——心理測試

這是一個心理測試項目，看看你是否一個樂觀的人。

磁碟雜誌中文字顯示原理剖析(二)

今期繼續講解螢幕顯示中文的原理。希望各位能領悟其中原理，它日能發揮應用。

視窗 HELLO

第二期介紹了一個SUPER HELLO ，用法是將各類型的FILE 分類顯示，但今期這個視窗HELLO 就極之強效，能將上捲的FILENAME 不斷的再顯示，而各項執行方式（RUN、LOAD等）只要按單字母即可。

BUBBLE SORTING 原理剖析及示範

BUBBLE SORTING ：排序在程式編寫中經常會用到，但大家是否清楚了解排序的工作原理？本文提供一個非常有條理的DEMO來詳細解釋BUBBLE 排序的工作原理。導師們最宜用來講解分析給學生聽。

圍棋遊戲

圍棋遊戲是供大家訓練思考力及精神定力的好媒介。各位可與電腦來比試。

愛在深秋

電腦音樂動畫程式——將一首名歌配上美麗的動畫。

第4期

低解像頁繪圖技巧

這篇自學文章詳細解釋了在低解像頁裏繪劃圖案的方法、原理，文章深入淺出，並在文末加上一個具體程式例子，要學習在低解像圖頁繪圖的方法，不能錯過本文。

心理測驗：你是否感受精神壓力

透過回答數十個問題，便可分析閣下現時所受到的精神壓力有多大，是否需要接受心理醫生輔導。

絲綢之路

絲綢之路是一首美妙的音樂，我們的作者將它配上生動的動畫，大家可安坐電腦機前欣賞這首美妙的樂章。

史諾比月曆

這個月曆能印出任何一個年份每月的月曆表，並配上一幅趣緻的史諾比漫畫，擁有印字機的讀者值得收藏。

拼字

這是考驗英文生字字滙能力的一個測驗，你敢接受挑戰嗎？

天降黏泥

這是一個編寫得非常精彩的低解像動畫遊戲，大家在娛樂之餘，不妨參考作者編寫本程式的技巧方法。

電子波子機

一個非常刺激的電子波子機遊戲，未知閣下能否打得一百萬分。

AMPERHELPER

這個工具程式盡量利用了&這個指令，有多個特別簡化的指令，值得大家收藏。

圖書館探險記

這是一個題材非常新穎的ADVENTURE遊戲，作者利用了圖書館編排圖書種類的系統，寫成了這個富有教育意義的遊戲。玩熟這遊戲，閣下在圖書館尋找資料就會快捷輕鬆得多。

電腦辭典

初接觸電腦的朋友，可能對電腦的各方面認識有限，我們特別編寫這個電腦名詞詞滙，將數十個常見的名詞加上註釋，令你對電腦的認識加深。

電子報告板

假如閣下有特別訊息要透過螢幕來告訴大家，不妨利用本程式。

西曆查農曆日子輸入方法之（三）

磁碟雜誌中文字顯示原理剖析（續完）

美國最新出版蘋果軟件

高登軟件流行榜

第5期

電子琴

能自己編寫任何心愛的歌曲，再重播出來。編曲方式簡易，效果動聽。

音樂動畫——似水流年

一首流行的熱門歌曲，配上美妙的動畫，湊成一個音樂動畫

小型迷宮——可大可小設計迷宮

一個保證能令人不肯放手的迷宮遊戲，迷宮的大小可隨意決定，難度亦有多層選擇。

文字加工遊戲

這是一個考驗你對英文生字認識多少的遊戲。玩法非常有趣，而且畫面構圖異常生動精彩。

英文打字訓練

這是一個由基本手法敎起的英文打字訓練輔助程序。對於有興趣學習快速而正確打字練習的同學是很好的幫手。

最高積分記錄程式

假如你是電腦遊戲高手，經常屢破紀錄，不妨用這個積分處理系統將之記錄下來。本程式能記錄超過一百個遊戲名字，而且可以將之順字母排序，功能良多。

SEOCTOR 編輯程式

這個工具程式比DISK ZAP好用得多。大家不妨比較一下。

貸款利率計算程式

這個工具能助你計算貸款應繳付的利息等問題。

電腦郵購表格

這個需要配合印字機使用的程式，可以輕易印出一份郵購貨品表格，適用於購買音樂節、藝術節門劵，或郵購電腦時代的產品。

蘋果老虎機

這個由低解像畫面組成的老虎機遊戲是一個難得的作品，趣味性很濃。

製造密碼程式

要破解密碼，需要長時間的分析及統計。本程式不單用來輕易地製造一些密碼考驗別人，而且亦可同時進行遊戲。

生命的海洋——心理測驗

這是一個心理測驗遊戲，測驗你對生命的看法。

誰是大偵探

誰是大偵探

要當一個出色的大偵探並非易事，不單要有精密的頭腦，客觀的分析，還要有很大的耐性。本遊戲可供最多4個人同時進行，看誰本領高，首先找出兇手。

新產品REVIEW

陽曆換算陰曆日子資料輸入之（五）完結篇

精彩的封面設計投稿

價廉物美的製作

JOYSTICK AUTO FIRE

朱志誠

相信大家都看過第十五期「電腦時代」中歡樂棒的AU—TOFIRE 製作。現在筆者介紹另一個同樣的製作，但這個「AUTOFIRE SYSTEM」的線路簡單得多，當然成本也很低，只需買一個555 IC （市值約$3.00）、SWITCH、電阻和VR便成。（如果想使成本更低，555 IC也不需要買）

AUTOFIRE就是自動連續按FIRE BUTTON，此工作可由一個振盪電路達成。起初筆者選用了一個晶體管振盪電路，但因輸出不足，所以便改用一個555 IC。當它和R1、R2、C1、VR 接駁後，就能產生一振盪信號，此信號足以使電腦檢知。現在介紹這製作的步驟：

(1)將圖一的線路接在一塊線路板上。爲了使體積和成本減低，VR是用微調的。單刀雙擲的SWITCH 用最小型的。

(2)將JOYSTICK脫離電腦，剪斷+5V和BUTTON 0 之間的電線。通常是橙色的。（如果想將BUT 1也有AUTO—FIRE，可將+5V和BUTTON 1之間的電線也剪斷。

(3)將圖一裏有「*」號的電線接至JOYSTICK 相對的接脚上。（例如+5接至JOYSTICK的+5V）

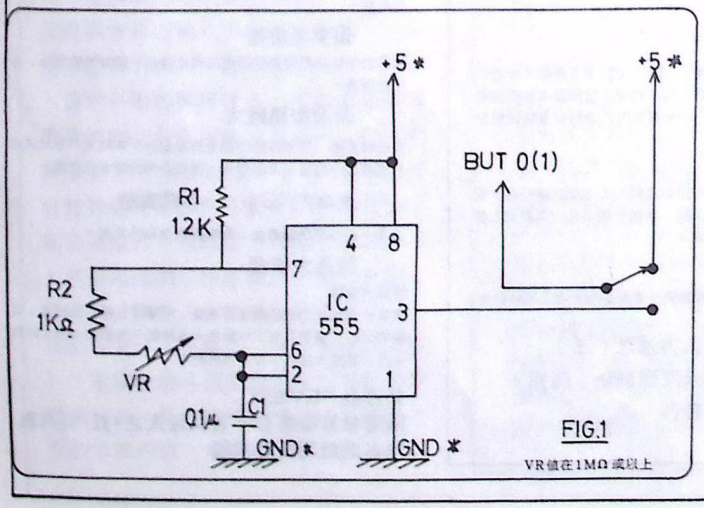
(4)在JOYSTICK 外殼鑽一個孔，安裝SWITCH，將線路板放在JOYSTICK內一角。蓋上JOYSTICK ，製作便完成了。

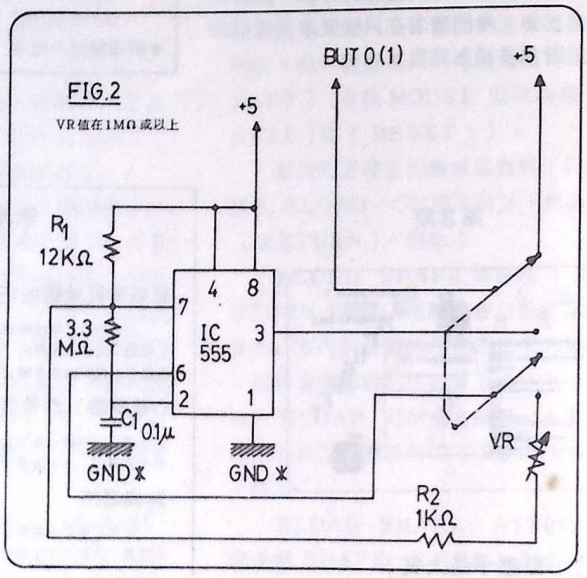
這個製作的好處就是整個SYSTEM 都放在JOYSTICK內，它可被插在任何一部APPLE內。

本人對電子方面不太熟悉，以上的線路是參照APPLE線路圖中控制顯示幕上閃爍字元的555IC振盪電路而造成的。正因爲此，我便想到借用APPLE內的555IC ，不需自己買。但是，線路的接駁比較困難，以下詳述之：

(1)找出電腦內位置「B3」的555 IC，小心拔出。

(2)將一條長電線（30號最佳）插在插座右面的第三脚上（如圖二，注意電腦的方向是左面電源、KEYBOARD向上）。





(3)然後將IC插回原處（注意IC上的小點和線路上其他IC的小點方向必須一致）。這條就是信號輸出線。

(4)將第二和第三條電線續在555IC前面的3.3MΩ 電阻兩端，（555IC前面有兩個電阻，較遠的一個才是3.3MΩ，參看圖三）注意不要將電線的銅絲接觸到其他地方，以免短路。

(5)在電腦內找一個+5V，將第四條電線接在此。（+5V在SPEAKER，SLOT，IC或GAME I／O等地方都可找到，大家可隨意選擇。）

(6)將四條電線穿出機外，（或放在平常輕易碰到的地方，）分別接至VR，SWITCH和電阻上。

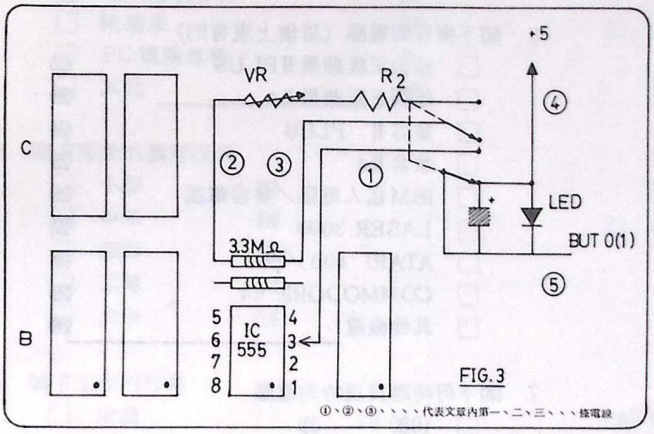
以上各步驟的詳細接線可參照圖三。

(7)最後就是要將JOYSTICK內+5V與BUTTON 0 間（通常是橙色）的電線剪斷，如果想BUTTON 1也AUTOFI—RE的話，可將+5V與BUTTON 1 間的電線也剪斷，再將第五條電線接在BUT 0 (1) 剛剪去的脚上。（爲美觀起見，這線可改爲接在GAME I／O的SW0(1)上）

這個線路跟上面的線路唯一不同的，就是前者多了一個3.3MΩ的電阻，和SWITCH改爲雙刀雙擲。這SWITCH 是用來選擇AUTO或MANUAL FIRE，而多出來的一「刀」是將VR和R2脫離原來的線路，使3.3MΩ 電阻兩端的阻值仍爲3.3MΩ，在MANUAL FIRE MODE 時，閃爍字元不會有快速閃動的現象。

在以上兩個製作中，都有一個VR，目的是用來調校AU—TOFIRE 的速度，因爲每個GAME讀取BUTTON 的頻

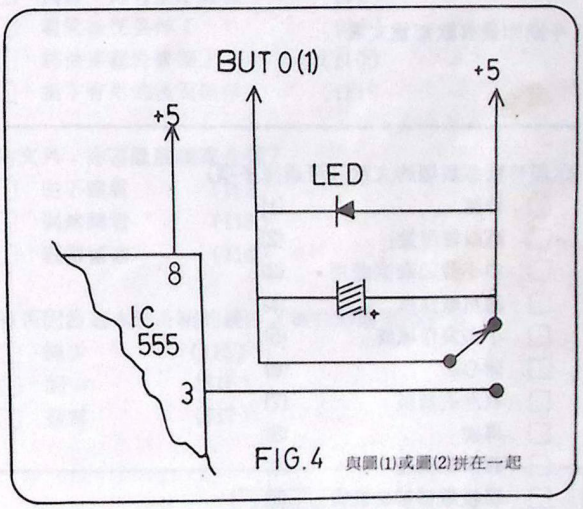
價廉物美的 JOYSTICK AUTOFIRE 製作



率（在一定的時間內讀取BUTTON 的次數）不同，所以要使AUTOFIRE 和讀取BUTTON 的頻率同步，才會以最高頻率「開火」，但只要兩者的頻率相差少許，可能結果「開火」速度會非常慢。如果將AUTOFIRE 的頻率加快到是一PROGRAM 讀取BUTTON頻率的20倍甚至更多時，結果對於讀取BUTTON頻率高的GAME（例如BANDITS）來講，的確快了很多，但「開火」的頻率非常不穩定，而對於讀取BUTTON頻率的GAME（例如ZAXXON)來講，「開火」速度更慢得要命，所以調校AUTOFIRE 的頻率就要視那種GAME而定。

另外一個問題又產生了。現在先簡述AUTOFIRE SY—STEM的工作原理。555IC不斷做出「ON」和「OFF」的信號到BUTTON，經過BUTTON 的另一脚才到電腦內部。BUTTON 0 (1)是用來將「ON」和「OFF」信號截斷與否，當AUTOFIRE SYSTEM 剛做完一個「ON」信號後，再變爲一個「OFF」信號時，如果此時按下BUTTON，那麼「OFF」信號便傳到電腦去，正因爲OFF，所以PRO—GRAM便以爲沒有按BUTTON，那麼便要等到AUTO—FIRE SYSTEM 再作出「ON」信號時，PROGRAM 才收到「ON」信號。如果AUTOFIRE 的頻率低的話，情形就很明顯。因此，在線路裏加了一個小電容量（例如0.33 μf）的電解質電容器。原理是將AUTOFIRE SYSTEM 做出來的信號的電儲起來，就算有以上的情形發生，電容器能及時放電來「開火」。此外，可將一個小型燈泡或LED 裝在線路上，顯示「ON」（燈泡亮起）和「OFF」(燈泡不亮）的狀態。圖四便是改良後的線路的一部份。可與圖一或圖二拼在一起。

此製作的成功率非常高，成本比第十五期陳一鴻君介紹的低，線路比上期的簡單，不失爲一個超值、簡易而實用的製作。■



妙趣CATALOG

DAVID

可能你會嘗試CALL－151，AD98L，你會見到很多JSR $FDED和JSR $AE42 和$AE2F，嘗試以EA（NOP）代之，將會有很多有趣的效果，例如

ADB3：EA EA EA （除去DISK TITLE）
ADC0：EA EA EA （除去VOL #）
ADE5：EA EA EA （除去LOCK CODE）
ADF9：EA EA EA （不印出檔類）
AE0B：EA EA EA （不印出檔長）
AE1B：EA EA EA （不印出檔名）
AE22：EA EA EA （除去回頭RETURN 在每一個檔名之後）

你估吓POKE 44592，41又是怎樣的CATALOG 呢？變化始終是由你自己發掘的好。■

讀者問卷調查

各位讀者：

電腦時代出版經已有十六期。我們不斷的努力，希望將電腦時代辦得盡善盡美，內容多元化發展。爲了進一步加強電腦時代的內容，務求能迎合大家的需要。我們現在進行一個問卷調查，希望大家能花兩分鐘的時間，回答下列問卷內的問題，我們將依據問卷的資料，來訂定未來的內容發展，請盡量忠實填寫。爲了感謝你們的幫忙，我們將會在收回來的問卷之中，抽出五十位朋友，致送一份紀念品，大家可在下列各項紀念品中選出一份：

- □ 任何一期電腦時代程式磁碟（填上第幾期）
- □ GPLE使用方法詳析
- □ MOCKING BOARD 應用技術

請在適當處加剔

1. 今期你最喜歡那個PROGRAM

_____

2. 今期你最喜歡那篇文章

_____

3. 你最喜歡那類型的文章（可選擇多項）
   - □ 特稿 (1)
   - □ 高級應用篇 (2)
   - □ 中小型工商業應用 (3)
   - □ 應用軟件庫 (4)
   - □ 小型製作專欄 (5)
   - □ 開心坊 (6)
   - □ 軟件縱橫談 (7)
   - □ 專欄 (8)
   - □ 蘋果工具箱 (9)
   - □ 學校電腦輔助教育 (10)
   - □ 兒童電腦輔助教育 (11)
   - □ 遊戲軟件評介 (12)
   - □ BACK UP 資料館 (13)
   - □ 精品店 (14)

4. 電腦時代應該增加些什麼內容？（請盡量發表意見）

_____

_____

_____

_____

5. 閣下的年齡：
   - □ 10～15 (15)
   - □ 16～20 (16)
   - □ 21～25 (17)
   - □ 26～30 (18)
   - □ 31～35 (19)
   - □ 36～40 (20)
   - □ 41以上 (21)

6. 閣下擁有的電腦（請填上現有的）
   - □ 原裝正版蘋果ⅡPLUS (22)
   - □ 原裝正版蘋果Ⅱe (23)
   - □ 兼容Ⅱ PLUS (24)
   - □ 兼容Ⅱe (25)
   - □ IBM私人電腦／兼容電腦 (26)
   - □ LASER 3000 (27)
   - □ ATARI 400／800 (28)
   - □ COMMODORE 64 (29)
   - □ 其他機種 _____ (30)

7. 閣下何時購買現時的電腦
   - □ 1980 (31)
   - □ 1981 (32)
   - □ 1982 (33)
   - □ 1983 (34)
   - □ 1984 (35)
   - □ 1985 (36)

8. 閣下有否計劃於未來一年內購買下列任何一種電腦
   - □ 原版Ⅱe (37)
   - □ IBM／兼容電腦 (38)
   - □ Ⅱc (39)
   - □ MACINTOSH (40)
   - □ 其他 _____ (41)

9. 閣下擁有什麼週邊設備
   - □ 磁碟機一部 (42)
   - □ 磁碟機二部或以上 (43)
   - □ 印字機（何種牌子）_____ (44)
   - □ 16K RAM咭 (45)
   - □ Z 80 咭 (46)
   - □ 80字行咭 (47)
   - □ TALK咭 (48)
   - □ 128K RAM咭 (49)
   - □ MOCKING BOARD (50)
   - □ 中文咭 (51)
   - □ 其他（MOUSE等）_____ (52)

10. 有沒有朋友／同學每期要求借閱電腦時代
    - □ 有 (53)
    - □ 沒有 (54)

11. 每期有否購買程式磁碟
    - □ 有 (55)
    - □ 沒有 (56)
    - □ 從朋友處COPY副本 (57)
    - □ 從別處COPY副本 (58)

12. 除電腦時代，閣下每月還購買那本電腦雜誌

- ☐ 微型電腦專刊 (59)
- ☐ 現代電子 (60)
- ☐ 電腦月刊 (61)
- ☐ 0011電腦月刊 (62)
- ☐ 蘋果用家專刊 (63)
- ☐ 純蘋果 (64)
- ☐ PC電腦專刊 (65)
- ☐ 其他 _____ (66)

13. 閣下接受的教育程度

- ☐ 小學 (67)
- ☐ 中學 (68)
- ☐ 預科 (69)
- ☐ 工專 (70)
- ☐ 大學 (71)

14. 閣下工作的行業

- ☐ 求學 (72)
- ☐ 電腦業 (73)
- ☐ 服務性行業 (74)
- ☐ 政府工 (75)
- ☐ 工程 (76)
- ☐ 其他 _____ (77)

15. 閣下將電腦利用來

- ☐ 家庭娛樂方面 (78)
- ☐ 家庭管理／財政 (79)
- ☐ 教育學習 (80)
- ☐ 個人興趣 (81)
- ☐ 商業管理 (82)

16. 閣下每週花多少時間於電腦之上

- ☐ 少於2個小時 (83)
- ☐ 2—6小時 (84)
- ☐ 7—11小時 (85)
- ☐ 12—16小時 (86)
- ☐ 17小時以上 (87)

17. 你否曾加入任何的APPLE USER GROUP

- ☐ 有 (88)
- ☐ 否 (89)

18. 你對自己的電腦系統感到滿意嗎？

- ☐ 滿意 (90)
- ☐ 不滿意 (91)
- ☐ 希望日後再加添更多硬件 (92)

19. 你有沒有購買外國電腦雜誌作參考

- ☐ 甚少 (93)
- ☐ 間中 (94)
- ☐ 經常 (95)

20. 個人每月收入（如仍依賴家庭供養無須填此欄）

- ☐ 2000～3000元 (96)
- ☐ 3000～4500元 (97)
- ☐ 4500～6000元 (98)
- ☐ 6000元以上 (99)

21. 個人每月能負担用於投資電腦的預算

- ☐ 少許50元 (100)
- ☐ 50～100元 (101)
- ☐ 100～300元 (102)
- ☐ 300元以上 (103)

22. 從何時開始，你歡喜閱讀電腦時代

- ☐ 看到一些非常實用的軟件 (104)
- ☐ 從市面各種電腦期刊中選擇發現本刊最能適合你 (105)
- ☐ 看到許多有參考價值的文章 (106)
- ☐ 剛買電腦，想找一本書參考 (107)

23. 每月收到電腦時代後

- ☐ 擺於一旁有空才翻看 (108)
- ☐ 看完後便丟掉了 (109)
- ☐ 將他珍藏於書架上 (110)
- ☐ 撕下有用的幾頁保存 (111)

24. 除內文外，你喜歡閱讀廣告嗎？

- ☐ 從不翻看 (112)
- ☐ 偶然翻看 (113)
- ☐ 經常留意 (114)

25. 你有否因看過本刊介紹的硬件／軟件而購買該產品

- ☐ 絕少 (115)
- ☐ 間中 (116)
- ☐ 經常 (117)

抽獎券

姓名：_____

電話：_____

地址：_____

_____

身份證號碼：_____

請將此抽獎券連同問卷同信封寄往九龍郵政信箱71193號 電腦時代收，信封請勿封口，並請在信封面註明「問卷調查」。

GRAPHIC SOLUTION
一套嶄新的動畫製作軟件

威利

市面上最近出現了一套設計頗爲新穎的動畫製作軟件，它就是ACCENT SOFTWARE所推出的TGS（THE GRA—PHIC SOLUTION）。TGS 除了設計標奇立異外，就連它的目標市場也與衆不同，一般的動畫製作軟件，如GRAP—HICS MAGICIAN，THE COMPLETE GRAPHIC SY—STEM等，都是強調自己在遊戲設計的優點，而ACCENT SOFTWARE却強調TGS 在商界的應用。

到底TGS 在商界有甚麼用途呢？我們可從以下簡單的例子裏得到答案。假如你是某一間公司的營業代表，而你正想向你的顧客介紹一種公司的新產品，那麼，你便會想，怎樣介紹這產品才會使顧客留下良好和深刻的印象呢？利用傳統的口述或死板板的圖片介紹，除了未能使顧客清楚瞭解產品外，還可能使顧客因煩悶而對產品失去興趣。在這種情形下，TGS 便是最佳的助手。你可將產品的資料和特點，通過TGS 彩色繽紛的動畫和適當的字幕顯示，清楚和有趣地介紹給你的顧客，這樣不但能使他們對你的產品有深入的瞭解，還可使他們留下深刻的印象。TGS 除了在產品的推廣有用外，當你要向上司報告公司的銷售狀況，市場調查或財政預算等時，TGS 都是很好的幫手。

筆者說TGS 設計新穎，是由於製作一套TGS 的動畫，就像在製作一套卡通片一樣。舉例說，你想利用TGS 來模仿一架穿梭機向上飛的情景，那麼你首先要利用TGS 的EDITOR 分別繪出一架穿梭機（動畫的主角）和它的背景，圖1a及1b ，然後將穿梭機放在背景的不同位置上，每到一個新的位置時，便可命令TGS 將那個畫面攝錄下來，當全部所需的畫面攝錄後，便可利用TGS 的放映機（PRO—JECTOR）將每一幅剛才攝製的畫面以高速播放出來，畫面的高速交替使我們感覺到穿梭機正向上飛去，圖1c—1e便是整套動畫的其中三幅畫面。由上面我們可以看出，每一個TGS 的畫面就等於卡通片的一格菲林，所有的畫面組合起來便是一套卡通片。（整套穿梭機起飛的動畫已收錄在磁碟月刊內，讀者有興趣的話，可自行修改或加入自己喜愛的圖案和動畫。）

TGS 所需的設備十分簡單，只需48K 的APPLE Ⅱ（或內置APPLESOFT ROM的APPLE Ⅱ PLUS）和一部磁碟機已足夠。TGS 整套軟件包含了三張磁碟，它們的名稱分別是PROGRAM，DEMO和SEQUENCES。

PROGRAM 磁碟內包含了TGS 的EDITORS，PROJE—CTOR 和其他有關的ROUTINES ，是製作和修改TGS 動畫不可缺少的磁碟。

DEMO和SEQUENCES 其實都是可有可無的磁碟。—DEMO 磁碟裏是一連串已製作好的TGS 示範動畫，從這些動畫中，你可看出TGS 在動畫製作各方面的功能。SEQU—ENCES 磁碟裏則是一系列的TUTORIALS，這些TUTO—RIALS 教你一些有關製作TGS 動畫的高級技巧，所以只是一些比較熟習TGS 的用家所需的。

另外，爲了使用家能清楚TGS 的原理、結構和操作方法，ACCENT SOFTWARE 也推出了TGS的MANUAL，這本MANUAL 對於TGS 的結構和操作方法的描述都十分詳盡，但對於TGS 原理（前文已述）的描述似乎略嫌不足

圖1a

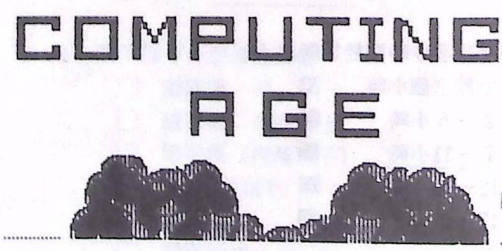


圖1b

圖1c

，這是MANUAL 比較失敗的地方。但總括而言，當用家把整本MANUAL 看過和實習過一次後，他已能把ＴＧＳ玩弄於掌上。

要製作一套ＴＧＳ的動畫，除了要懂得ＴＧＳ的原理外，也必須熟習ＴＧＳ的八個不同狀態（MODE），這八個狀態分別是MENU，LO—RES，HI—RES，TEXT，LINE，SHOW，SPEED和CIRCLE 。要進入某一個狀態，必需在另一個特定的狀態裏按下適當的功能鍵才可，圖２便是狀態與狀態之間的進出關係。每一個狀態在ＴＧＳ的動畫製作中都有自己一定的用途，它們並且各自擁有一組屬於自己的功能鍵。

一般的動畫製作軟件，都只是佔用APPLE 的高解像畫面，而ＴＧＳ却同時佔用高解像和低解像兩個畫面。唯一在低解像畫面使用的ＴＧＳ狀態便是LO—RES ，其餘 TEXT，LINE，CIRCLE，HI—RES，SHOW和SPEED 狀態都是在高解像畫面使用的，而MENU 則是在正常的文字幕裏使用。

LO—RES 狀態的作用十分大，除了可用來設計動畫的主角外，還可將高解像畫面局部放大，搬到低解像畫面裏進行編輯。TEXT 狀態容許你直接由鍵盤鍵入ＴＧＳ內置的文字到畫面任何位置。LINE 狀態則是繪畫直綫和直綫圖形的最佳助手。而在CIRCLE 狀態裏，你只要按一適當的功能鍵，便可在畫面任何位置畫出一個圓形。TEXT，LINE 和CI—RCLE 都主要用於背景的製作。當主角和背景製作後，便可在HI—RES 狀態裏將它們組合起來，並按下適當的功能鍵將畫面攝成動畫的其中一格菲林。當全部畫面攝製後，便可利用SHOW 狀態來試播一次；播放的速度可用SPEED 狀態來調校，直到一切滿意後，便可到MENU 狀態將動畫SA—VE 到磁碟去。

站在純動畫製作的觀點來看，ＴＧＳ 的確比起其他的同類軟件有不少優勝之處：

1）ＴＧＳ 能夠使用MACRO 指令。假如你有一串功能鍵需要作重複性的執行，那麼你便可將這串功能鍵定義成爲一個MACRO 指令，當你執行這個MACRO 指令時，便等於執行那串功能鍵一樣。

2）ＴＧＳ 的內置文字能直接由鍵盤鍵入高解像畫面的任何一個位置上。

3）ＴＧＳ 動畫的播放速度可作隨意調校，就算連某一幅畫面與它下一幅畫面的交替速度也可作隨意的更改。

4）高解像畫面可作局部性的放大，並搬到低解像畫面進行編輯，待編輯完畢後，便可搬回高解像畫面的原處去。

5）ＴＧＳ 內置有畫圓形的功能，這使弧形圖像製作不再是一件難事。

6）由於ＴＧＳ 的動畫是以一格一格的菲林組成，而其他同類軟件的動畫則是整體性和不可分解的，所以在修改一套已完成的動畫時，ＴＧＳ 就變得容易得多。在ＴＧＳ裏，只要將要修改的菲林從動畫中抽出進行編輯便可，而在其他同類軟件中，往往要將整套動畫拋掉，從新製造另外一套新的動畫，這是非常不值和費時的。

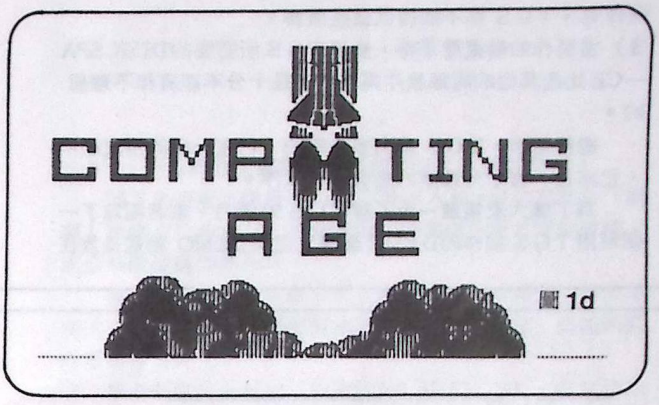


圖 1d

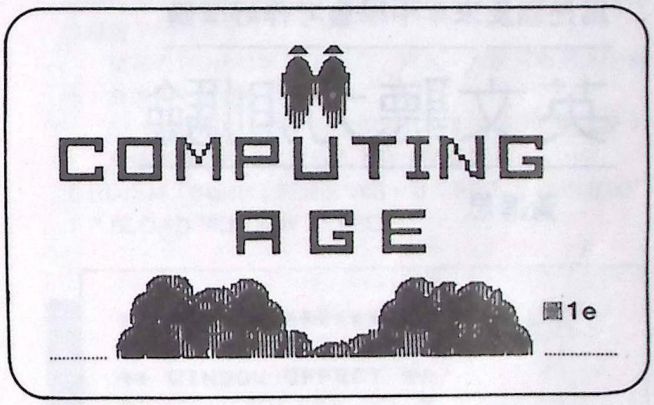


圖1e

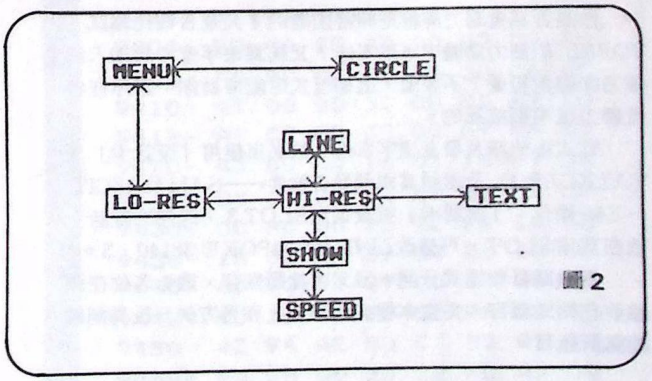


圖 2

俗語說有利必有弊，當筆者用了ＴＧＳ一段時間後，便發覺ＴＧＳ也存在着一些缺點：

1）由於ＴＧＳ 的每個狀態都有自己的一組功能鍵；而每組功能鍵的數目都不少，但ＴＧＳ 的畫面上却沒有任何功能鍵的提示，這迫使用家要牢記所有的ＴＧＳ 功能鍵，或者在每次使用ＴＧＳ時都要拿着MANUAL 查看，這是十分不方便的

2）ＴＧＳ 在背景製作的功能似乎略嫌不足。像用家希望將他畫好的圖形塡上顏色，或想改用操縱桿來控制一切的畫圖

一套嶄新的動畫製作軟件

動作等，TGS 都不能提供這些服務。

3）當製作的動畫簡單時，使用TGS所需要的DISK SPACE比起其他的同類軟件爲多，這是十分不經濟和不理想的。

總括而言，TGS 是利多於弊的，它在一般的用家眼中，已算是一套十分理想的動畫製作軟件。

爲了讓大家能進一步了解TGS的威力，筆者編寫了一個利用TGS製作的DEMO動畫。這個DEMO 動畫是放在今期的程式磁碟中，共有多個FILE，各位只要RUN CONTROLLER即可。

(1) TGSDEMO1・S
(2) TGSDEMO2・S
(3) CONTROLLER
(4) PROJECTOR
(5) TGS TEXT

■

爲托福及來年中學會考作好準備

英文聽力測驗

黃博思

你是否感覺聽力測驗是非常困難的？又是否在托福試TOFEL 的聽力測驗中表現不佳？又或爲來年會考新加入的聽力測驗而担憂？不要緊，這個程式定能帮助你，讓你在各種聽力試中顯威風的。

程式由於涉及發音讀字部份，故必須使用「說話卡」（TALKCARD）及連同其兩個發音軟件——SAM及RECITER 操作。「說話卡」被設定到SLOT3，若你的說話卡放在其他SLOT，可修改LINE 185的POKE 38140，3。

測驗題目和程式分開，以文字檔型儲存，避免各位在測驗前已熟知題目，失去本程式的用意，亦可方便日後爲測驗編寫新題目。

操作指引

程式開始時先表示說話卡在某（1—6）SLOT 中，要求你按RETURN 繼續程式，若你並無說話卡在該SLOT，便應按ESC退出程式。

程式會由磁碟中讀入SAM及RECITER兩個發聲系統，然後讀入題目「ATP L.1」，由於要連續三個FILES，筆者提議使用一些快DOS（如DIVERSE DOS）執行，節省時間。

程式接着印出按鍵指示，用左右箭頭選擇答案，和聲音示範。聲音效果亦是用AMPENSAND ROUTINE 寫成。然後在三下號角聲後，測驗便正式開始，你會聽到一段說話，然後在螢光幕上看到四欄選擇，和托福試一樣。在螢光幕下方會看到時計和得分。時計是在倒數着的，在倒數未完之前你可改變答案選擇，時限到了便响電鈴，讀出下一題。答對一題加十分，否則倒扣十分，同時會告訴你正確的答案。筆者在ATP L.1 檔案中輸入了二十題，問完題目便印出分數，結束程式。

本程式必須使用 TALK CARD

各位同學留意，本程式必須要使用一張TALK CARD，假如沒有此咭，則本程式並不能執行。而且，由於本程式在執行時，會呼叫附隨TALK CARD的軟件——SAM 及REC—ITER ，故必須將此兩個程式載入同一張磁碟上。此外，本文共有兩個程式，一個是APPLESOFT主程式，LISTE—NING TEST，另一個是文字檔案「ATP L.1」。這裏並不把程式列表刊出，但各位可在今期的程式磁碟中找到這兩個檔案。

難度的改良

讀句速度和音調由LINE 310的P和SP控制，用有限亂數做出。基本上較高的音調和較快的速度會提高問題的難度，各位不妨按程度降低P和SP的值。

時限由LINE 180 定出，各位亦可按自己的反應挑戰，基本上半分鐘是極充裕的了。

最後，祝各位同學在各項考試中有理想的表現！■

將視窗顯示功能注入 II PLUS

翻看過介紹MAC文章的讀者都知道MAC能用視窗來顯示多層次的程序。大家對於「視窗顯示」功能應該不會感到陌生，許多高級的商業性軟件都內置這種「用家友善」的功能。

過去，APPLE Ⅱ由於受到記憶體的限制，在視窗顯示功能方面是難以想像的，不過，今日閣下大可以引以自豪的對同伴說：「我的APPLE 亦能做到視窗顯示功能！」

筆者設計的這個滙編語言程式能夠讓你產生最多三個層次的視窗顯示，這項功能所需要的最低條件只是一部磁碟機及48K 記憶量的主機，閣下只要在40字行顯示模式下便能做到。

當你開設一個「視窗」在文字幕時，在視窗週界範圍內的文字能任意捲動，而在「視窗」底下的文字幕則會「凍結」。當這個「視窗」被移去後，在於底下的文字幕又會再次活躍起來。當我們加上第二個「視窗」或第三個「視窗」時，先前底下的一個「視窗」或文字幕亦同樣受到「凍結」，不能活動，直至上面一個「視窗」被移去後，方才能「活動」。本程式能做到這項功能其原理筆者稍後再詳細解釋。

三個「視窗」的大小由 3 欄× 3 行至40欄 ×23行（正常40行文字幕是40 ×24）。但「視窗」週邊佔用了一行，換言之，實際能於「視窗」內顯示的文字幕寬度及高度都要減 2。

「視窗」的位置可任意在螢幕上設定，只要不超出下列兩個原則：

(1)「視窗」的濶度加上「視窗」開始的水平位置不超過41欄。

(2)「視窗」的高度加上「視窗」開始的高度位置之和不能超過24行。

假如有任何一個「視窗」設定超出這兩個限制，則會導致程序中斷。緊記著，本程序最多只能製造三個「視窗」。假若超出這個數目，程序會出現兩次"ERR"字樣及响起鐘聲，然後繼續執行原先的程式。

APPLE 本身其實早已內置有捲動視窗功能——整頁（40欄×24行）文字幕捲動。在監督程式內已有 4 個記憶位置管理著文字幕捲動的範圍——左邊32，右邊33，上邊34，下邊35。當我們PEEK這幾個位置時，其值爲32等於 0，33等於40，34等於 0，35等於24。這些值與文字幕範圍值相類型。故此，若我們POKE 入不同值於這幾個位置上，則會局限了文字幕某個範圍的捲動功能。假如各位不相信，可直接打入下列一句看看：

POKE 33，20=POKE 32，10：POKE 34，5：POKE 35，20〔RETURN〕。

跟著打入CATALOG或LOAD入一個FILE，然後LIST。所有的捲動會局限於某個POKE入值的範圍。

閣下大可嘗試採用不同的POKE入值，但切記勿令「視窗」範圍大於40×24，這會導致記憶體中的程式受到破壞，或影响磁碟機的操作。

要恢復原來的文字幕，只要用TEXT指令即可。任何不同的POKE入值都會回復原本面貌，而在「視窗」範圍內的內容都會消失。

筆者所編寫的程式（WINDOW EFFECT），就是利用這個原理。它能讓你產生最多三個的視窗重叠而不會破壞每個視窗下的內容。

要利用WINDOW EFFECT 程式，大家可照列表(1)打入，然後用下列指令存入磁碟：

BSAVE WINDOW EFFECT，A$9400，L$1CF。

然後在你的程式開首打入下列幾句：

```
0 HIMEM：34815：POKE 768，0：PRINT CHR$(4)
；"BLOAD WINDOW EFFECT"
```

表(1)

```
**********************
*                    *
** WINDOW EFFECT **
*                    *
**********************
9400- 20 4A FF AD 00 03 C9 00
9408- F0 5C CE 00 03 20 C6 95
9410- A9 00 8D 3C 00 8D 42 00
9418- A9 CF 8D 3E 00 A9 04 8D
9420- 43 00 AD 16 03 18 69 88
9428- 8D 3D 00 18 69 03 8D 3F
9430- 00 A0 00 20 2C FE 18 AD
9438- 16 03 AA 69 04 8D 18 03
9440- A0 00 BD 09 03 99 20 00
9448- EC 18 03 F0 05 E8 C8 4C
9450- 42 94 AE 00 03 BD 05 03
9458- 8D 25 00 20 22 FC A9 00
9460- 85 24 20 3F FF 60 20 2D
9468- FF 20 2D FF 20 3F FF 60
9470- 20 39 FB A2 00 BD 88 94
9478- C9 AA F0 07 20 F0 FD E8
9480- 4C 75 94 A2 35 4C 12 D4
9488- D7 C9 CE C4 CF D7 BA AD
9490- AA 20 4A FF AD 00 03 C9
9498- 03 F0 CB 18 AD 01 03 6D
94A0- 02 03 E9 28 10 CA AD 02
94A8- 03 38 E9 03 30 C2 18 AD
```

```
94B0- 03 03 6D 04 03 E9 17 10
94B8- B7 AD 04 03 38 E9 03 30
94C0- AF 20 C6 95 18 AD 03 03
94C8- 6D 04 03 8D 17 03 AE 00
94D0- 03 AD 25 00 9D 05 03 AD
94D8- 32 00 8D 19 03 AD F3 00
94E0- 8D 1A 03 18 AD 16 03 AA
94E8- 69 04 8D 18 03 A0 00 B9
94F0- 20 00 9D 09 03 EC 18 03
94F8- F0 05 C8 E8 4C EF 94 A9
9500- 00 8D 3C 00 8D 42 00 A9
9508- CF 8D 3E 00 A9 04 8D 3D
9510- 00 18 69 03 8D 3F 00 AD
9518- 16 03 18 69 88 8D 43 00
9520- A0 00 20 2C FE 20 80 FE
9528- A9 01 8D 21 00 AD 01 03
9530- 8D 20 00 AD 03 03 8D 22
9538- 00 AD 17 03 8D 23 00 20
9540- 58 FC AE 04 03 20 BD 95
9548- 18 AD 01 03 6D 02 03 AA
9550- CA 8E 20 00 20 58 FC AE
9558- 04 03 20 BD 95 AE 01 03
9560- E8 8E 20 00 AE 02 03 CA
9568- CA 8E 21 00 20 58 FC 20
9570- BD 95 CE 20 00 EE 21 00
9578- EE 21 00 AE 17 03 CA 8E
9580- 22 00 20 58 FC E8 E8 8E
9588- 23 00 AE 02 03 CA 20 BD
9590- 95 CE 21 00 CE 21 00 EE
9598- 20 00 AE 03 03 E8 8E 22
95A0- 00 CE 23 00 CE 23 00 EE
95A8- 00 03 20 58 FC AD 19 03
95B0- 8D 32 00 AD 1A 03 8D F3
95B8- 00 20 3F FF 60 A9 20 20
95C0- F0 FD CA 10 FA 60 AD 00
95C8- 03 0A 0A 8D 16 03 60 AA
```

要在螢幕上設定一個視窗，首先要必須設定四個記憶位址值：

POKE 769，A
POKE 770，B
POKE 771，C
POKE 772，D

A、B、C、D 代表文字幕四邊的範圍（順序爲左、右、上、下）。A 的值可由 0 至37，B 值由 3 至40，C 值由 0 至20，D 值由 3 至23。

打入CALL 38033 開啓「視窗」。打入CALL 37888 則可移去「視窗」。爲了令大家能熟習本程式的用法，筆者寫了一個示範程式（WINDOW EFFECT DEMO），程式可見列表(2)。

現在扯回討論主程式的兩大子程式（開設視窗及移去視窗）的原理及其記憶位址。讓大家可深入了解主程式的工作過程。

開設視窗子程式：

位址 $9494—$94C0：檢查錯誤子程序，視窗不能小於 3×3 或大於40×23，並且不能多過三個視窗。

位址 $94C1—$94FE：監督各項正常功能確保正常。可以保持現有視窗，游標位置，及在使用「移去視窗子程式時，使用PRINT指令」。

位址 $94FF—$9524 在新一個視窗產生前將文字幕內容存入高位置記憶區。螢幕 1 存於 $8800—$8C 00 ，螢幕 2 存於 $8C00—$9000，螢幕 3 存於 $9000—$9400 。（本文主程式是存於 $9400—$9600）

位址 $9525—$95A6 劃出視窗範圍，及設定範圍內的捲動。

位址 $95A7—$95BC在產生新一個視窗之前，RESTORE PRINT模態。

以下爲移去視窗子程式的各個記憶位址：

位址 $9403—$9409 檢查錯誤，假如沒有視窗，則會出現 ERROR 1訊息。

位址 $940A—$9435 RESTORE被SAVE 起來的文字幕內容。

位址 $9437—$9451 RESTORE 反白了的文字幕邊界。

位址 $9452—$9465 RESTORE 新視窗之前的游標垂直位置及交回程式執行權。■

表(2)

```
0   REM ****************
1   REM *                *
2   REM * WINDOW EFFECT*
3   REM *      DEMO      *
4   REM *BY   COMPUHOUSE*
5   REM *                *
9   REM ****************
11   HIMEM: 34815: POKE 768,0: PRINT
      CHR$ (13) +  CHR$ (4);"BLOA
     D WINDOW EFFECT"
15   TEXT : HOME : LIST
20   DATA 0,40,10,10,15,20,0,23,4,
      27,3,14
30   RESTORE : LIST : FOR I = 1 TO
      3
40   FOR MEMLOC = 769 TO 772
50   READ WSIZE: POKE MEMLOC,WSIZE

60   NEXT : CALL 38033: REM  PUSH

70   LIST : LIST : LIST : LIST
80   NEXT
90   FOR I = 1 TO 3
100   FOR T = 1 TO 1000: NEXT
110   CALL 37888: LIST : REM  POP
120   NEXT : FOR T = 1 TO 1000: NEXT
      : GOTO 30
```

配合單軌 INIT 程式使用法

改良版40 軌磁碟改造器

KENNY JIM

筆者在閱讀十三期之〔40軌磁碟改造器〕，曾嘗試運作，發覺其未能達到預期效果。同時在研究第十一期之SING—LE TRACK INIT（見第11期P.41）後，發覺在搬移後具更大彈性，能配合BASIC程式操作，故特編寫此一個驅動程式，（見列表①），並配合第十一期的SINGLE TRACK INIT 程式以達至不影响磁碟內容而增加五條TRACKS 之目的。大家可將本程式視爲40軌磁碟改造器的修訂版。

爲配合BASIC驅動程式，SINGLE TRACK INIT 需要作以下修改，讀者可於輸入SINGLE TRACK INIT （編者註，大家可於第十一期的雜誌上找到原本程式，本原始碼程式是用LISA 2.5 編寫）後，打入：

```
I 24
 24  ;
 25  △ORG △$ 1000
 26  ;
 27  ∧E
I 95
 95  ;
 96  △ORG △$ 1090
 97  ;
 98  △LDA △#$ 11
 99  △LDY △#$ 00
100 △JMP △L03D 9
101 △END
102 ∧E
D  103，104
```

筆者經多次試驗後，此程式效果非常理想，而且能在作少量修改後，配合各種需要。

驅動程式解說：

100　測試是否已讀入TRACK INIT ，若否，便將之從S 6，D1 中讀入。

在此只測試其中兩個BYTES ，錯誤機會爲 $\frac{1}{65536}$

備註：

KEY：

△：SPACE

∧E ：CTRL—E

其中95—100 行是利用RWTS 來讀寫VTOC，若不願冒險，可把IF STATEMENT刪去，但每次RUN時均需把有TRACK INIT的磁碟放進S6，D1。

160—170 設立IOB 表。

利用BASIC建立IOB 可方便修改。

180—190 提示放進磁碟。若希望選擇DRIVE 2，可作以下修改：

```
180 VTAB 10：INPUT "DRIVE NO：" ；DR$
182 DR=VAR（DR$）：IF DR<>1 AND DR <>
    2 THEN 180
184 POKE IOB+2，DR
```

200—250 讀入VTOC

260—310 修改VTOC並將之寫進磁碟。

若希望同時釋放TRACK 2之11個空SECTORS 可加入：

```
295 POKE BUFFER+64，255：POKE BUFFER +
65，224
```

但雖小心，程式不會理會這11 SECTORS 是否有資料。

320—340 用TRACK INIT來INIT TRACK 35至TRACK 39。同時將正在INITIATING的TRACK NO. 顯示出來。

因一般磁碟均是35 TRACKS FORMAT 的，此程式只修改此等磁碟和36TRACKS 的，若希望修改其他軌數的磁碟，可修改或刪除250行的IF STATEMENT。如希望INIT第41條TRACK ，可將290行的40改爲41，300行和340行的39改爲40。

相信讀者亦發覺此程式其實可INIT 任意數目的磁軌，只要作適當刪改便可。

至於其他發展，則留待讀者自己發掘以適應需求。例如：INIT一張41 TRACKS 的資料磁碟（總容量爲164 KB）。

本文共有兩個程式：(1)INIT TRACK 35 TO 39； (2) TRACK INIT。前者是APPLESOFT 驅動程式，後者是用LISA 2.5 寫成的原程式。各位在打入時，如果沒有LI—SA 2.5 則可照左手邊的機械碼打入，並用下列指令存入磁碟：

BSAVE TRACK INIT，A$ 1000，L $ 98

大家可在今期的程式磁碟中找到INIT TRACK 35 TO 39及TRACK INIT的經ASSEMBLY 後的目的程式。■

改良版 40 軌磁碟改造器

```
10  REM ************************
20  REM *                      *
30  REM * INIT TRACKS 35 TO 39 *
40  REM * FOR A 35-TRACKS DISK *
50  REM *                      *
60  REM *    KENNY JIM S.K.    *
70  REM *  20TH JANUARY, 1985. *
80  REM *                      *
90  REM ************************
100  IF  PEEK (4096) <  > 32 OR  PEEK
     (4240) <  > 169 THEN  PRINT
      CHR$ (4);"BLOAD SINGLE TRAC
     K INIT,$6,D1"
110 IOB = 4352:BUFFER = 8192
120  TEXT : HOME : VTAB 3: HTAB 1
     2: INVERSE : SPEED= 150: FOR
     I = 1 TO 16: PRINT  MID$ ("4
     0 TRACKS EXTEND",I,1); CHR$
     (7);: NEXT : NORMAL : SPEED=
     255: PRINT
130  REM
140  REM *** IOB
150  REM
160  RESTORE : FOR I = IOB TO IOB
      + 20: READ V: POKE I,V: NEXT

170  DATA 1,96,1,0,17,0,17,17,0,3
     2,0,0,1,0,0,96,1,0,1,239,216

180  VTAB 10: PRINT "PUT IN THE D
     ISK IN DRIVE 1"
190  PRINT : PRINT "AND PRESS A K
     EY.";: GET KEY$: PRINT
200  REM
210  REM *** READ VTOC
220  REM
230  CALL 4240
240 TR =  PEEK (BUFFER + 52):VOL =
      PEEK (BUFFER + 6)
250  IF TR > 36 THEN  PRINT : PRINT
     "IT IS NOT A 35 TRACKS DISK!
     ?"; CHR$ (7): GOTO 360
260  REM
270  REM *** SAVE VTOC
280  REM
290  POKE BUFFER + 52,40
300  FOR I = TR TO 39: POKE BUFFE
     R + 56 + I * 4,255: POKE BUF
     FER + 57 + I * 4,255: NEXT
310  POKE IOB + 12,2: CALL 4240
315  REM
320  REM *** INIT TRACK 35 TO 39
325  REM
330  PRINT : PRINT : PRINT "TRACK
      : ";
340  FOR I = TR TO 39: POKE 2,I: POKE
     3,VOL: HTAB 9: PRINT I: CALL
     4096: NEXT
350  PRINT  CHR$ (7): PRINT "INIT
     IATION FINISH!"
360  PRINT : PRINT "DO YOU WANT T
     O INIT ANOTHER DISK?";
370  GET KEY$: IF KEY$ = "Y" OR K
     EY$ = "y" THEN 120
380  IF KEY$ <  > "N" AND KEY$ <
      > "n" THEN 370
390  PRINT KEY$
```

```
0800                1  ;
0003                2  L03      EPZ $0003
0002                3  L02      EPZ $0002
0001                4  L01      EPZ $0001
0000                5  L00      EPZ $0000
002D                6  L2D      EPZ $002D
003E                7  L3E      EPZ $003E
0045                8  L45      EPZ $0045
0044                9  L44      EPZ $0044
0041               10  L41      EPZ $0041
0048               11  L48      EPZ $0048
03D9               12  L03D9    EQU $03D9
03E3               13  L03E3    EQU $03E3
0578               14  L0578    EQU $0578
B8DC               15  LB8DC    EQU $B8DC
B944               16  LB944    EQU $B944
BB00               17  LBB00    EQU $BB00
BBFF               18  LBBFF    EQU $BBFF
BF0D               19  LBF0D    EQU $BF0D
C089               20  LC089    EQU $C089
C088               21  LC088    EQU $C088
FDDA               22  LFDDA    EQU $FDDA
FDED               23  LFDED    EQU $FDED
0800               24  ;
1000               25           ORG $1000
1000               26  ;
1000 20 E3 03      27           JSR L03E3
1003 84 00         28           STY L00
1005 85 01         29           STA L01
1007 A5 02         30           LDA L02
1009 A0 04         31           LDY #$04
100B 91 00         32           STA (L00),Y
100D A9 00         33           LDA #$00
100F A0 0C         34           LDY #$0C
1011 91 00         35           STA (L00),Y
1013 A9 00         36           LDA #$00
1015 A0 03         37           LDY #$03
1017 91 00         38           STA (L00),Y
1019 20 E3 03      39           JSR L03E3
101C 20 D9 03      40           JSR L03D9
101F BD 89 C0      41           LDA LC089,X
1022 A5 02         42           LDA L02
1024 85 44         43           STA L44
1026 A5 03         44           LDA L03
1028 85 41         45           STA L41
102A A9 AA         46           LDA #$AA
102C 85 3E         47           STA L3E
102E A9 28         48           LDA #$28
1030 85 45         49           STA L45
1032 A0 56         50           LDY #$56
1034 A9 00         51           LDA #$00
1036               52  L0836:
1036 99 FF BB      53           STA LBBFF,Y
1039 88            54           DEY
103A D0 FA         55           BNE L0836
103C               56  L083C:
103C 99 00 BB      57           STA LBB00,Y
103F 88            58           DEY
1040 D0 FA         59           BNE L083C
1042 20 0D BF      60           JSR LBF0D
1045 A9 08         61           LDA #$08
1047 B0 19         62           BCS L0862
1049 A9 30         63           LDA #$30
104B 8D 78 05      64           STA L0578
104E               65  L084E:
104E 38            66           SEC
104F CE 78 05      67           DEC L0578
1052 F0 0E         68           BEQ L0862
1054 20 44 B9      69           JSR LB944
1057 B0 F5         70           BCS L084E
1059 A5 2D         71           LDA L2D
105B D0 F1         72           BNE L084E
105D 20 DC B8      73           JSR LB8DC
1060 90 1F         74           BCC L0881
1062               75  L0862:
1062 A0 0D         76           LDY #$0D
1064 91 00         77           STA (L00),Y
1066 A9 87         78           LDA #$87
1068 20 ED FD      79           JSR LFDED
106B A9 D2         80           LDA #$D2
106D 20 ED FD      81           JSR LFDED
1070 A9 C3         82           LDA #$C3
1072 20 ED FD      83           JSR LFDED
1075 A9 BD         84           LDA #$BD
1077 20 ED FD      85           JSR LFDED
107A A0 0D         86           LDY #$0D
107C B1 00         87           LDA (L00),Y
107E 20 DA FD      88           JSR LFDDA
1081               89  L0881:
1081 BD 88 C0      90           LDA LC088,X
1084 A9 00         91           LDA #$00
1086 85 48         92           STA L48
1088 60            93           RTS
1089               94  ;
1090               95           ORG $1090
1090               96  ;
1090 A9 11         97           LDA #$11
1092 A0 00         98           LDY #$00
1094 4C D9 03      99           JMP L03D9
1097              100           END

***** END OF ASSEMBLY
```

把損壞的磁軌鎖住

SAMUEL LUI

參閱第十五期"將含有I／O ERROR的SECTOR除去"一文後，心想既然有一部APPLE 在手，何不改爲電腦操作，於是便嘗試寫個程式來代替人手。由於半年前收到了一盒品質甚差的磁碟，也實在需要整理一下以便能物盡其用，也就着手以LOCKSMITH 5.0 以找出損壞的磁碟部份，但嘗試過了LOCKSMITH 5.0 的所有程式後，最多只能找出損壞的磁軌，而不能指出損壞的扇區。雖不能如願，能夠鎖住損毀的磁軌，也可救回九成以上的磁碟空間了，於是便寫成以下名爲"TRACK LOCKER"的程式了。

使用時，先用 LOCKSMITH 5.0 找出損壞的磁軌號碼，再INITIALIZE 有問題的磁碟，然後RUN本程式，待螢幕出現"PUT IN DEFECTIVE DISK IN DRIVE 1"時，便可換入有問題的磁碟，按入雖要鎖上的磁軌號碼便全部完成。整個程式相當USER FRIENDLY ，沒有特別需要註釋之處。

由於LOCKSMITH 5.0 所指示出損壞的磁軌號碼以十六進制表示，爲方便起見，本程式按入磁軌號碼亦是十六進制，使用時便省却一番功夫，雖然這樣一來，運作速度會拖慢了，但整個程式因爲相當簡短，全部運作也只是半秒之間。

程式說明：整個程式所佔最大部份便是偵錯和轉爲十進制的一段。因爲BASIC 語言必定以十進制編寫的，所以轉爲十進制是必需的步驟。句20至30用以載入RWTS副程式，因此而可以用最簡單的方法讀寫任何一條磁軌。句60—150爲偵錯及轉爲十進制。句160爲讀出VTOC（參閱第十五期）。句170爲修改資料，把需要鎖上的磁軌在DIRECTORY上改爲已被佔有。句180則爲將VTOC 重寫。

又因爲本程式相當簡短快捷，所以並沒有在結尾時再回頭。用家如有多過一條磁軌需要鎖上，則可以再RUN 一次，時間無大分別。美中不足的是手上沒有任何程式可以測知損壞的扇區，將來再有的話，則可改寫爲一個鎖上個別扇區的程式，則可用回的磁碟空間又再增加了。■

```
1   REM  ************************
2   REM  *                      *
3   REM  *    TRACK    LOCKER   *
4   REM  *                      *
5   REM  *     BY SAMUEL LUI    *
6   REM  *                      *
7   REM  *      APRIL 1985      *
8   REM  *                      *
9   REM  ************************
10   REM
15   REM
20   FOR X = 896 TO 896 + 30: READ
      D: POKE X,D: NEXT
30   DATA  169,3,160,138,32,217,3,
      96,0,0,1,96,1,0,17,0,153,3,0
      ,32,0,0,1,0,0,96,1,0,1,239,2
      16
40   HOME : VTAB 3
45   PRINT "PUT IN DEFECTIVE DISK
      IN DRIVE 1,": PRINT
50   INPUT "ENTER TRACK NO. TO BE
      LOCKED (HEX): ";TK$: PRINT
60 TK$ =  RIGHT$ (TK$,2): IF TK$ <
      > "11" THEN 80
70   PRINT "TRACK 11 CANNOT BE LOC
      KED, ": PRINT "PLEASE RE-ENT
      ER": PRINT : GOTO 50
80 TEN =  ASC ( LEFT$ (TK$,1)) -
      48
90   IF TEN >  = 0 AND TEN < 3 THEN
      110
100   PRINT "PLEASE ENTER TRACK NO
      . FROM 03 TO 22": PRINT : GOTO
      50
110 UNIT =  ASC ( RIGHT$ (TK$,1))
      - 48
120   IF UNIT > 22 OR UNIT < 0 THEN
      100
130   IF UNIT < 10 THEN 150
140 UNIT = UNIT - 7
150 TK = TEN * 16 + UNIT
160   CALL 896
170   POKE TK * 4 + 8248,0: POKE T
      K * 4 + 8249,0: POKE 918,2
180   CALL 896
190   PRINT : PRINT "TRACK ";TK$;"
       LOCKED,"
200   PRINT : PRINT "    ...JOB DON
      E..."
```

機械語言怎樣控制磁碟機馬達活動

磁碟機的控制是經由存取一系列的控制暫存器所達到。每一塊CONTROLLER 咭都可以控制兩個磁碟機的運作，但在一個時間之內只能控制其中一部磁碟機，經由16個控制暫存器（REGISTER）來操作。

當磁碟機的CONTROLLER 咭插入其中一個SLOT時，在APPLE 記憶體中它們的相對暫存器位址相對由$C0N0至$C0NF，其中N等於SLOT號碼加上8，以插在SLOT 6的CONTROLLER 爲例，控制暫存器的位址是由$C0E0至$C0EF，這些控制位址排列如下表。

以上的寄存器位址都可以提供表上的功能，方法是只需將該功能的寄存器位址上內容載入到累加器便行，但上表的暫存器位址並未包括控制磁碟機的讀寫，因爲讀／寫是由其它特別的位址來控制的。主要的是，在控制磁碟機的過程之中，我們只需針對馬達與步進馬達（STEPPER MOTOR）的運作，至於磁頭的讀／寫是另一回事，我們暫時可以不理。因此，我們爲了要控制馬達和步進馬達，我們必須載入$C0N0～$C0NF 的內容到累加器之上，在BASIC 模式之中我們可以用PEEK和POKE指令達成，以機械語言來處理時，我們可以用LDA或BIT 這些等效指令來載入這些功能的位址內容，以技術名詞來說，這些處理方式叫作REFER—ENCING。

C及D暫存器內容（內置在位址$C0EC及$C0ED 之中，$C0EC及$C0ED是假定CONTROLLER 咭插在SLOT 6之上的），不論在任何情形之下都要將它們的內容載入累加器或由累加器中寫回到它們之上，因爲它們是用來選擇CONTROLLER咭作爲讀／寫之用的。

啟動磁碟機馬達的方法

當我們載入表中MOTOR．ON（$C0N9）這個位址的內容到累加器上時，CONTROLLER 咭上的TTL便會將電流通往磁碟機的馬達，使磁碟機上紅色的LED燈亮起並且令馬達運轉，各位不妨試將CONTROLLER 咭放在SL—OT 6 之上（一般我們都插入CONTROLLER咭在SLOT 6），鍵入下列指令：

```
POKE 49385，0      （磁碟機啓動）
CALL—151           （磁碟機仍在轉動）
C0E8               （中止馬達轉動，LED 熄掉）
```

以上的49385其實相當於$C0E9，是MOTOR・ON的暫存器。這個暫存器是任何磁碟機操作過程中的第一個必須使用的內容。

選擇第一部或第二部磁碟機

在上表中的A及B暫存器（位址是$C0NA及$C0NB）的功效一如電視機的頻道選擇，只要選擇它們其中之一，

ADDR.（位址）	標　記	功　能
$C0N0	MAG0・OFF	關閉步進馬達的第 0 相位。
$C0N1	MAG0・ON	開著步進馬達的第 0 相位。
$C0N2	MAG1・OFF	關閉步進馬達的第 1 相位。
$C0N3	MAG1・ON	開著步進馬達的第 1 相位。
$C0N4	MAG2・OFF	關閉步進馬達的第 2 相位。
$C0N5	MAG2・ON	開著步進馬達的第 2 相位。
$C0N6	MAG3・OFF	關閉步進馬達的第 3 相位。
$C0N7	MAG3・ON	開著步進馬達的第 3 相位。
$C0N8	MOTOR・OFF	關閉磁碟機馬達。
$C0N9	MOTOR・ON	開著磁碟機馬達。
$C0NA	SEL・DRV1	令CONTROLLER選擇DRIVE 1 。
$C0NB	SEL・DRV2	令CONTROLLER選擇DRIVE 2 。
$C0NC	IO 1	控制I／O的選通脈冲（STROBE LATCH）。
$C0ND	IO 2	載入數據脈冲（DATA LATCH）。
$C0NE	IO 3	令CONTROLLER咭開啓閘門預備輸入。
$C0NF	IO 4	令CONTROLLER咭開啓閘門預備輸出。

機械語言怎樣控制磁碟機馬達活動

將它們其中之一的內容例如B的載入累加器內，便會令CONTROLLER咭的功能全部選擇到第二部磁碟機之上。同樣原理，A暫存器會令所有功能全部選擇第一部磁碟機執行。以下列表1的程式是示範首先開啓磁碟機馬達，然後快速地由第一部磁碟機選擇第二部磁碟機，再由第二部選回第一部，閣下不妨試一下。

若果閣下有兩部磁碟機，而CONTROLLER 咭是插在SLOT 6之上時，磁碟機的LED便會輪流亮起。假如閣下只有一部磁碟機時，它的LED會閃亮，好像訊號燈一般，原因是列表1的程式首先啓動磁碟機馬達（LDA $C0E9），再輪流選擇磁碟機。各位可以按入任何鍵來中止程式執行。在$2FF位址上的內容是延遲時間之用，換句話說是控制馬達啓動後維持的時間，若果閣下將它降低，例如按入

2FF：20 〔RETURN〕

300 G 〔RETURN〕

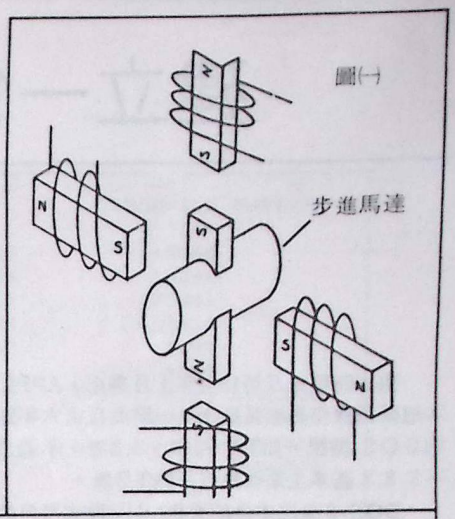
若果你有兩部磁碟機的話，在感覺上會覺得兩部機是一齊亮著的。

步進馬達的控制方法

到現在爲止，我要和各位研究磁碟機上的另一個馬達，稱爲步進馬達（STEPPER MOTOR）的控制方法。各位應該知道磁碟機之中是有兩個馬達的，第一個馬達當我們採用MOTOR·ON（$C0N9）暫存器內容時便會啓動，以一個穩定的速度（300RPM）轉動，直到工作完畢爲止它才會停止，它的控制方法在上面我們已說明過。

```
10  HOME : INPUT "SLOT=>";SL: PRINT
    : INPUT "DRIVE=>";DR
20  PRINT : INPUT "CURRENT PHASE=
    >";CP: PRINT : INPUT "DESTIN
    ATION PHASE=>";DP
30        AD = 49152 + 128 + SL *
    16: POKE AD + 9,0: POKE AD +
    9 + DR,0
40  IF DP = CP THEN 110
50  IF DP ,> CP THEN 80
60  FOR A = CP * 2 + 1 TO DP * 2 STEP
     - 1
70  POKE AD + A -  INT (A / 8) *
    8,0: NEXT : GOTO 110
80  FOR A = CP * 2 + 1 TO DP * 2 +
    1 STEP 2
90  POKE AD + A -  INT (A / 8) *
    8,0
100  POKE  - 1 + AD + A -  INT (A
     / 8) * 8,0: NEXT
110  POKE AD + 8,0: END
```

表2



圖(一)

表1

```
02FF-   C0                DELAY TIME
0300-   AD E9 C0          LDA   $C0E9
0303-   AD EA C0          LDA   $C0EA
0306-   AD FF 02          LDA   $02FF
0309-   20 A8 FC          JSR   $FCA8
030C-   AD EB C0          LDA   $C0EB
030F-   AD FF 02          LDA   $02FF
0312-   20 A8 FC          JSR   $FCA8
0315-   AD 00 C0          LDA   $C000
0318-   10 E9             BPL   $0303
031A-   AD E8 C0          LDA   $C0E8
031D-   AD 10 C0          LDA   $C010
0320-   60                RTS
```

第二個馬達是一個十分特別的步進馬達，它的結構圖可以簡化爲如圖1所示。當我們載入$C0N0至$C0N7 的內容時，便會使它的四個相位中的其中一個通電流產生磁場，吸引步進馬達的軸上安裝的永久磁石到該相應的相位去，這就是我們所稱的循跡效應（ALIGNMENT）。

磁頭在磁碟機上的移動是由以上的步進馬達來控制的，當然這項工作最好是由DOS來負責（因爲它有著準確的延遲網路來配合轉動馬達來讀寫資料），但我們可以採用以下列表2的BASIC 程式作爲實驗，它可以將磁頭移到任何位置上去（包括半磁軌HALF TRACK）。

各位在執行列表2的程式之前，最好先緊記DOS是將磁碟劃分爲70個相位（35磁軌） （80個相位則爲40磁軌），其中每一個相位都相對應一個步進馬達的位置，我們稱之爲半磁軌。通常DOS在讀寫資料之時是採用偶數相位的（EVEN PHASE，如0，2，4 等），故此第二個相位我們通常都叫作第一個磁軌TRACK 1，第4個相位叫TRACK 2，第68個相位叫TRACK 34。而奇數相位ODD PHASE 便叫做半磁軌（HALF TRACK）。

當你使用這個移動磁頭的程式時，假如你不清楚現在磁頭是在那一個相位的話，你可以先輸入80作爲現在相位CURRENT PHASES，然後再輸入0作爲目的相位DESTINATION PHASE，這樣你便會將磁頭重校回到TRACK 0位置之上，這時的相位是零。各位若果對以上資料感興趣的話可以參看由DOH WORTH所著的BENEATH APPLE DOS，在這本書內你會找到更多的對你有用的內容。■

建立一個更好的DOS系統

PRODOS

由1978年6月至1984年1月爲止，APPLE Ⅱ系列的基本磁碟機操作系統是DOS ，雖然在這六年間曾經推出不同的DOS 型號，如3.1，3.2，3.3等，不過最後推出的DO—S 3.3 基本上是改良以往的3.2版。

DOS 3.3原本是給予BASIC程式用戶的磁碟機操作系統，但却逐漸演變爲連機械語言程式的編寫亦使用它。由於DOS 3.3 是APPLE 系列的操作系統中最便宜者（比較PASCAL 及CP／M需要額外週邊硬件），因此它亦是最普遍被使用的系統。

1984年1月，APPLE推出PRODOS——一個給予AP—PLE 系列的全新作業系統。這個PRODOS 是準備用來取代DOS 3.3 的磁碟機作業系統，現在若果閣下購買APP—LE磁碟機的話，便可以免費得到PRODOS的系統磁碟，取代以往的DOS 3.3 系統磁碟。

PRODOS 的面世已有一年多了，我們早在第四期及第五期發表過PRODOS 的指令REVIEW，之後筆者在第7至13期之間介紹了PRODOS 的使用方法（第7期），磁碟的格式（第8期），啓動過程及記憶體之使用技術（第9期），PRODOS 系統對記憶體的頁數使用及檔案系統的認識（第10期），在APPLE 相融的電腦上執行PRODOS（第11期），在PRODOS上BASIC解釋器的機械碼進入點（第12期），使用PRODOS的PREFIX 指令技巧（第13期）。這些內容都是針對PRODOS 的SYSTEM磁碟。

記得在去年五月，PRODOS 首次在坊間出現。當時只有兩隻磁碟，第一是系統磁碟，第二是公用程式磁碟而MA—NUAL 則只有很簡的PRODOS USER GUIDE 的影印本，此外甚麼資料也沒有了。直到同年8月坊間推出BAS—IC PROGRAMMING IN PRODOS ，這本書是指導初學者如何在BASIC程式之內利用PRODOS 來操作磁碟機的資料存取。書本上是採用隨同的程式磁碟來作說明，不過很遺憾這面磁碟直到同年12月才出現，使用戶不能在第一時間內充份發揮這本書的好處。另一方面，PRODOS的公用程式（UTILITY） 磁碟雖然早就有了，不過却沒有MANU—AL，使其上的MLI ，EXERCISER，EDASM 等程序得物無所用，直到今時今日這本MANUAL 還沒有出現過。就這樣，PRODOS 的熱潮冷却了，不過PRODOS在事實上是一個設計得很好的作業系統，在美國出品的APPLE 最新軟件都大多數以PRODOS 爲作業系統，而APPLE 公司亦有意使APPLE Ⅱ系列永遠生存下去——APPLE ⅡC 推出即可以證明這一點，因爲這個原故，我們若果不掌握PRO—DOS 的話實在是一件很遺憾的事。但首先我們要知道甚麼才是完整的PRODOS 作業系統？

APPLE 公司推出PRODOS 的時候，將PRODOS 系統賣給兩個層次的用戶，第一個層次的用戶是初學者，APP—LE 公司爲他們準備了①PRODOS USER'S KIT ，這包括一本MANUAL 及公用程式磁碟，售價是40美元，② PRO DOS BASIC PROGRAMMER'S KIT ，包括了一本MANUAL 及兩隻磁碟，主題是針對使用BASIC 配合PRO—DOS 系統作資料存取。

第二個層次的用戶是高級使用者，APPLE 公司爲他們準備了 ① PRODOS TECHNICAL REFERENCE PACKAGE，這包括一本叫作PRODOS TECHNICAL RE—FERENCE MANUAL的書，售價25美元。 ② PROD—OS ASSEMBLER TOOLS ，這包括一隻磁碟，磁碟上包括了組譯器及除錯程式，連同MANUAL售價50美元。

以上的4個套件便構成了完整的PRODOS 系統，到目前爲止，坊間還沒有ASSEMBLER TOOLS的MANUAL。

假如我們要發展機械語言程式的話會熱切地需求它，雖然現正處於有鎖沒有鎖匙的環境下，不過這本MANUAL 是針對較高階的使用者，在一般情況下我們都可以先在DOS 3.3 系統中以LISA或MERLIN BIC MAC 等組合器預先組合，再利用PRODOS 系統磁碟上的CONVERT系統程式將文本檔案及目的檔（二進制檔案），由DOS 3.3格式轉換爲PRODOS 的檔案，這是一個折衷的辦法，雖然較爲麻煩，但事實證明十分有效。以下我以第14期的「高解像圖魔術師」這個程式來作一個實例說明：

「高解像圖魔術師」共有兩個程式，一個是用BASIC 編寫的控制程式，在我們出品的程式磁碟上（CA14A） 名叫 HI—RES MAGIC DRIVER
第二個是主程式，採用BIG MAC MACRO組合器編寫的滙編程式，原程式在第14期P.25～31 頁上，經過BIG MAC 組合之後，產生目的程式檔，名叫HI—RES MAGIC。

現在我們要將這個程式在PRODOS 系統上執行，因此我們先啓動PRODOS 的SYSTEM磁碟，在主目錄上選擇DOS ↔PRODOS CONVERSION 這項功能（按入C鍵），你會見到螢幕上出現一個叫做CONVERT MENU 目錄，第二行顯示的DIRECTION 是轉換方向，這時候應該是DOS 3.3 S6，D2—→PRODOS，你可以把第14A 程式磁碟放入第二部磁碟機上，因爲現在我們要將DOS 3.3 的檔案轉換爲PRODOS 檔案，因此我們不需按入R 鍵來改變轉換方向，之後你需要把PRODOS 的工作碟放入第一部磁碟機上〔這面工作碟必須預先經過PRODOS FORMAT過，如何FO—RMAT PRODOS 磁碟可以參考PRODOS USER GUIDE 的第33頁～52頁。〕

到現在爲止，我們假定你已經把DOS 磁碟放入DRIVE 2，PRODOS 磁碟放入在DRIVE 1 上，這時你只要按入P 鍵來設定PRODOS磁碟的PREFIX，我們按S鍵以SLOT 及DRIVE 來設定（這樣做最簡單），我們選擇S6，D1 。

建立一個更好的DOS系統 —— PRODOS

沒有MANUAL 的 UTILITY DISK目錄

```
 NAME            TYPE  BLOCKS  MODIFIED          CREATED          ENDFILE SUBTYPE

*PRODOS          SYS       31   1-JAN-84  0:00  <NO DATE>          15360
*BASIC.SYSTEM    SYS       21  15-NOV-83  0:00  <NO DATE>          10240
*CONVERT         SYS       42   1-NOV-83  0:00  <NO DATE>          20481
*FILER           SYS       51  <NO DATE>        <NO DATE>          25600
 HELLO           BAS        7  16-NOV-83 15:57  <NO DATE>           2699
*EDASM.SYSTEM    SYS        9  14-DEC-83 15:32  <NO DATE>           4095
*EDASM.ED        BIN       17  14-DEC-83 15:32  <NO DATE>           8191 A=$3000
*EDASM.ASM       BIN       29  14-DEC-83 15:26  <NO DATE>          14079 A=$3000
*RBOOT           BIN        1  14-DEC-83 15:35  <NO DATE>            410 A=$0218
*RLOAD           BIN        3  14-DEC-83 15:36  <NO DATE>            661 A=$0800
*EXERCISER       SYS       16  17-OCT-83  0:00  <NO DATE>           7385
*BUGBYTER        BIN       16  15-DEC-83  0:00  <NO DATE>           7177 A=$2000
*FORMATTER       TXT        3  20-DEC-83  0:00  20-DEC-83  0:00     764 R=     0
*APA             BIN       10  11-NOV-83  7:37  <NO DATE>          4608 A=$2000
*BYE             BAS        1  16-NOV-83 15:55  <NO DATE>           158
*ABLE            BIN        4  21-NOV-83 11:10  <NO DATE>          1060 A=$2000
 BELL            TXT        1   5-FEB-84  0:00   5-FEB-84  0:00     292 R=     0
 CONVERTING.RAND BAS        5  <NO DATE>        <NO DATE>          1623
 OBJ.BELL        BIN        1  15-MAY-84  0:00  15-MAY-84  0:00      43 A=$5000
 B               BIN        1   9-JUL-84  0:00   9-JUL-84  0:00      43 A=$5000
 TE              TXT        1   8-JUN-84  0:00   8-JUN-84  0:00     299 R=     0

BLOCKS FREE:     3     BLOCKS USED:  277     TOTAL BLOCKS:  280
```

假如閣下希望採用PATHNAME來設定的話，只要按入P，再鍵入 VOLUME 名／PATHNAME（參考USER GUIDE P.105至P.119 頁可得更詳細說明）。

你亦可以在這時候按D鍵來輸入日期，之後我們按T鍵，按〔RETURN〕，螢幕便會出現DOS磁碟的目錄，我們可以用左右箭咀鍵來選擇，在第14期的程式磁碟HI—RES MAGIC DRIVER及HI—RES MAGIC是在第一及第二個檔位上，因此我們只要按"SPACE BAR"再用右箭咀鍵移到第二個檔名上按SPACE BAR，之後你只要按入RETURN鍵，CONVERT系統便會執行轉換工作。完成後按0離開，載入BASIC SYSTEM後便可執行。

若果你按入CAT命令來看PRODOS 檔案時，你會見到兩個檔案，一個名叫HI.RES. MAGIC.DR ，另一個叫HI.RES. MAGIC，第一個是BASIC檔，第二個是二進制檔，它們名字和DOS上的都有了改變，因此我們要修改載入程式，你可以按入LOAD HI.RES. MAGIC. DR 〔RE—TURN〕，將原來第190行改爲190 PRINT D$；"BL—OAD HI. RES. MAGIC"，之後把程式SAVE回碟上即可，使用高解像圖魔術師的方法和一切功能都沒有改變，但存取速度方面却快上8倍，這是PRODOS的一項優點。

到目前爲止，我已經向各位說明如何選擇完整的PRO—DOS系統，亦討論過一些不使用PRODOS ASSEMBL—ER 的折衷辦法，這種方法不但充份發揮CONVERT 系統程序功能，亦說明了如何將DOS 3.3 上的有用程式轉到PRODOS 系統上執行。不過，你首先得明白，甚麼環境下應該採用PRODOS ？

甚麼環境下應該採用 PRODOS

① 從技術觀點而言，PRODOS對磁碟機的存取速度比—DOS 3.3快8倍，因此在資料庫管理，或存取次數衆多的場合中使用PRODOS是明智之舉。

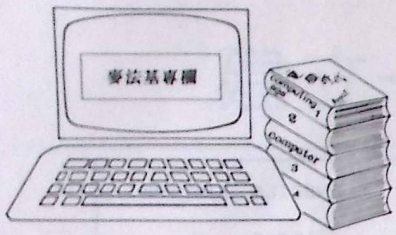
② 連接硬磁碟機的環境下，PRODOS 可以將高達32MB的磁碟儲存量視爲一個邏輯單元處理，使硬磁碟機的性能盡展所長，尤其現在5MB 的硬磁碟機售價僅爲二千多元，價價錢方面十分有吸引力，因此若果連接硬磁碟機的話，儘可能應當選用PRODOS。 DOS 3.3 對硬磁碟機必須先劃分爲一個個的邏輯區域，對5MB 硬磁碟機來說是劃分爲40個軟性磁碟，從而限制了檔案大小，使得許多空間被浪費了。

③ PRODOS 可以在系統內加插一些週邊咭的驅動程式，當你加裝週邊裝置（如BUFFER PRINTER CARD），CLOCK CARD，MOCKING BOARD等）時，可以不需對整個系統作大修改便能夠將所需的裝置驅動程式加到去PRODOS之內。

④ APPLE IIe的程式作業系統使用PRODOS 可以自動設置RAM DISK，方便程式使用更大記憶體。

總結

由於 DOS 3.3 已經推出有一段很長的時間，而有關於它的技術資料我們都已充份掌握，比較起PRODOS的技術來說，還距離一段時日，加上我們有許多應用程式都是針對DOS 3.3 的特性來設計，因此即使明知DOS 3.3 存有很多缺憾，也必須繼續耗下去，但有一點可以肯定的是 —— DOS 3.3與PRODOS仍會在APPLE 作業系統中佔一個重要角色，而PRODOS將會是APPLE作業系統的明日之星。 ■



掀開蘋果高解像畫面的神秘面紗（二）

LOOK UP TABLE查表法

前一期筆者曾經提及有關高解像度的HPOSN程式，不過HPOSN雖然解決了大部份高解像度位址問題，不過顯然太緩慢了，所以除了一部份不要求速度的程式外，很多程式設計者都不採用。

倘若要求速度的話，則必須應用LOOK UP TABLE查表法。不但是畫面應用查表法，很多其他方面亦須求查表速度。其實查表亦是和HPOSN一樣原理，當我們要在畫面上繪圖時，必須首先找到該位置的最左面位址。所不同的地方是HPOSN把位址計算出來，而查表法是預先把位址存在表上，當有需要時只要到表內找出來便可以了。很明顯地畫面上所有位址都是固定的，所以HPOSN的計算不是太麻煩嗎？

查表方法以間接索引（INDEX INDIRECT ADDRESS—ING）進行。由於位址包括高低位元，所以需一對指位器作爲位址暫存，至於間接索引如何應用，在上一次已經談過了。

現在，筆者以一個不斷移動的銀河星群作爲查表法的應用。相信大家都知道，很多太空遊戲都會把群星移動作爲遊戲的背景，使遊戲更加有動感。試想一下銀河群星可能包括數拾至一百粒星星在不斷向左右或上下移動，倘若用HPOS—N作爲繪畫程式，則顯然十分緩慢，不用說還要加上其他物體的移動了。星星向上下移動還可以避免水平顏色問題，不過很多時都需要作水平移動，所以問題更加難於應付。

這一期所列出的星群移動程式（共三個程式），純粹是針對查表法而編寫，所以筆者暫時避開位元（BIT）的移動，下一期將會提供一個完整位元移動的程式。事實上，如果畫面只要求所有星星作同一速度左右移動的話，則今期的程式已經足夠應付了。假如希望星星各自有不同速度向不同方向移動，則下一期的程式便可提供所有應用方法。

銀河星群原理十分簡單，首先在畫面上以亂數繪畫出一定數目的星星，當然星星的定義只是一個小點而已。假設我們希望所有星星以同一速度向左移動，當星星到達左面後必須從右面再出現繼續移動。你有想過如何繪畫星星和使它們移動嗎？當然你可以HPLOT一點，在舊地方消圖，然後在新的地方繪畫上新的一點，當星星到達左面邊緣時，再在左面畫上一點，如此類推達到移動效果。

這個原理一百份之一百正確，所不同的是不能再應用H—PLOT，HCOLOR之類的指令，就算以機械語言的速度，假如你應用以上的指令，則效果會令你覺得星星有如一群螞蟻在爬行而已。首先解決繪畫星星的問題，請鍵入HGR：POKE 8192，640你會發覺畫面左上角有一點星星顯現出來，我們可以明白到除了HGR是清除畫面之外，POKE 8192，64便是以繪畫上一點星星了。這樣的方法畫一點星星十分容易，因爲64的二進數是01000000，這個1便是星星了，不過怎樣畫出一群星星呢？請再鍵入以下一段小程式：

```
 5  HGR
10  N=8192
20  FOR I=1 TO 50:R= INT (RND(1)*100) :M
    =N+R:POKE M,64:N=M:NEXT
```

然後RUN這個程式。這時畫面上不是有50點不同位置的星星嗎？

明白以上繪畫星星的原理後，可以着手編寫一個星群移動的程式了。參看LISTING #1的組合語言，第34行的—GETADR是找出起始位址部份，YVERTH和YVERTL（LISTING #3）是一個預先造好的畫面位址表，事先以—PIL的檔案存於$9200位址上，請留意YVERTL，Y的Y—REGISTER是垂直位置座標值，只要存入一個需要的數值，則GETADR會從畫面位址表找出左面起始位址，然後把該位址低高位元存入HIRESL和HIRESH一對指位器內。

在未呼叫這個移動程式之前，我們首先要把星星所有位置告訴移動程式。這裏筆者要指出的是我們萬萬不能直接和高解像畫面上繪畫和消圖。因爲在畫面上並非只有星星存在，你可能會有一些太空船或怪物之類的圖形在移動，而我們亦要測試星星的雙單位置。移動一粒星星需要把該星星的BIT位置向右移作爲測試該位元組（BYTE）是單數或是雙數，下面會談到，不過倘若直接在畫面上繪畫星星，它的一BIT位置勢必被其他移動的物體改變。由於以上原因，我們另外設立一個星星位置分佈表，該表由$6000開始，參看L—ISTING #2的程式，第30行至50行便是以亂數產生星星位置分別把水平位元組和垂直數值存入24576至24628之間。

再查看LISTING #1的第41行STAR MOVE，SVERT是星星垂直位置，因爲每粒星星位置不同，所以首先由第59行的程式把要移動那一粒的垂直和水平位元組存入SVERT和HORIZ內，這時從GETADR找出左面起始位置，加上HORIZ便是這粒星星的實際位址，第44行作出消圖，只要把零存入該位址便可把星星消去了。第46行的Y—REG現時是水平值，測試它是否零，若是則星星已經到達左面界限，需要把該值改爲$27，則星星會改由右面界限移動。倘若未到左面界限，則由第53行開始把星星向左移一個位元組便可。

以上程序由第59行的STAR程式負責把每一粒星星重覆工作直至20粒星星移動完畢爲止。

當運行本程式時，螢幕上星星移動極快，所以LISTING #2的第70行要把運行減慢，否則速度超過應有效果。不過在眞正應用時，由於有其他物體及多種運算，無須再把程式減慢了。

這個程式可作爲參考之用，並非完善，因爲仍然未加上

LOOK UP TABLE 查表法

雙單位元組測試及位元的影响，所以畫面上星星有叠影出現。下一期會刊出一個完善的銀河星群程式。

很明顯地這個程式已經應用到兩個查表方法。一個是畫面位置查表，而另一個是星星位置的查表。星星的位置表是隨着星星移動而變動着，所以LISTING ＃1 的第67行要把星星移動新位置的水平部份重新放回表內，讓下一次移動查表時數值亦跟着變動，這是和畫面位置查表有所不同。

下次再談！■

```
: A S M                                              LISTING #1
                   1     ***********************************
                   2     *                                 *
                   3     *                                 *
                   4     *   BY PAUL F.K. MAK              *
                   5     *   14TH MARCH, 1985              *
                   6     *                                 *
                   7     ***********************************
                   8     *
                   9               ORG    $4000
4000: 4C 4E 40    10               JMP    START
                  11     *
                  12     *
                  13     *
                  14     *
                  15     *
                  16     XBYTE     EQU    $0         ;BYTE POSITION FOR DRAW
                  17     XBIT      EQU    $1         ;BIT POSTION FOR DRAW
                  18     VERT      EQU    $2         ;VERT POSTION FOR DRAW
                  19     HIRESL    EQU    $26        ;ADDRESS LO POINTER
                  20     HIRESH    EQU    $27        ;ADDRESS HI POINTER
                  21     STTEMP    EQU    $FD        ;TEMPORARY VARIABLE
                  22     SVERT     EQU    $FE        ;VERT POSTN VARIABLE
                  23     HORIZ     EQU    $FF        ;HORIZ POSTN VARIABLE
                  24     *
                  25     *
                  26     *
                  27     STARBYTE EQU    $6000      ;STAR BYTE POSTN TABLE
                  28     STARVERT EQU    $6020      ;STAR VERT POSTN TABLE
                  29     YVERTH    EQU    $9200      ;LOOK UP TABLE HIGH
                  30     YVERTL    EQU    $92C0      ;LOOK UP TABLE LOW
                  31     *
                  32     *
                  33     *
4003: B9 C0 92    34     GETADR    LDA    YVERTL,Y   ;LOOK UP LO BYTE OF LINE
4006: 85 26       35               STA    HIRESL     ;STORE TO POINTER
4008: B9 00 92    36               LDA    YVERTH,Y   ;LOOK UP HI BYTE OF LINE
400B: 85 27       37               STA    HIRESH     ;STORE TO POINTER
400D: 60          38               RTS
                  39     *
                  40     *
400E: A4 FE       41     STARMOVE LDY    SVERT      ;STAR VERT POSTN
4010: 20 03 40    42               JSR    GETADR     ;FIND LEFTMOST ADDRESS
4013: A4 FF       43               LDY    HORIZ      ;ADD HORIZ POSTN
4015: A9 00       44               LDA    #$00       ;ERASE STAR
4017: 91 26       45               STA    (HIRESL),Y ;ON GRAPHIC
4019: C0 00       46               CPY    #$00       ;STAR REACHES LEFT?
401B: D0 08       47               BNE    STARMOVE1  ;NO, MOVE STAR
401D: A0 27       48               LDY    #$27       ;YES, DRAW ON RIGHTMOST
401F: 84 FF       49               STY    HORIZ      ;STORE BACK TO VARIABLE
4021: A9 40       50               LDA    #$40       ;DRAW ON GRAPHIC
4023: 91 26       51               STA    (HIRESL),Y ;POSTION
4025: 4C 2F 40    52               JMP    STARMOVE2  ;RETURN
4028: 88          53     STARMOVE1DEY              ;SHIFT LEFT BY ONE BYTE
4029: 84 FF       54               STY    HORIZ      ;RESET VARIABLE
402B: A9 40       55               LDA    #$40
402D: 91 26       56               STA    (HIRESL),Y
402F: 60          57     STARMOVE2RTS
                  58     *
4030: A2 00       59     STAR      LDX    #$00       ;FIRST STAR
4032: 86 FD       60     STAR1     STX    STTEMP     ;STORE TO VARIABLE
4034: BD 00 60    61               LDA    STARBYTE,X ;LOOKUP BYTE POSTN
4037: 85 FF       62               STA    HORIZ      ;STORE TO TEMP VARIABLE
4039: BD 20 60    63               LDA    STARVERT,X ;LOOKUP VERT POSTN TABLE
403C: 85 FE       64               STA    SVERT      ;STORE TO TEMP VARIABLE
403E: 20 0E 40    65               JSR    STARMOVE   ;MOVE STAR
```

LOOK UP TABLE 查表法

```
4041: A6 FD      66                LDX  STTEMP      ;RESET X-REGISTER
4043: A5 FF      67                LDA  HORIZ       ;NEW HORIZ POSTN OF STAR
4045: 9D 00 60   68                STA  STARBYTE,X  ;STORE TO BYTE TABLE
4048: E8         69                INX              ;NEXT STAR
4049: E0 14      70                CPX  #$14        ;FINISHED ALL STARS?
404B: 90 E5      71                BLT  STAR1       ;NO, MOVE ANOTHER ONE
404D: 60         72                RTS              ;YES, RETURN
                 73   *
                 74   *
404E: 20 30 40   75   START    JSR  STAR
4051: 60         76             RTS

--End assembly--

82 bytes

Errors: 0

Symbol table - alPhabetical order:

   GETADR  =$4003      HIRESH  =$27        HIRESL  =$26        HORIZ   =$FF
   STAR    =$4030      STAR1   =$4032      STARBYTE=$6000      STARMOVE=$400E
   STARMOVE1=$4028      STARMOVE2=$402F      START   =$404E      STARVERT=$6020
   STTEMP  =$FD        SVERT   =$FE     ?  VERT    =$02     ?  XBIT    =$01
?  XBYTE   =$00        YVERTH  =$9200      YVERTL  =$92C0

Symbol table - numerical order:

?  XBYTE   =$00     ?  XBIT    =$01     ?  VERT    =$02        HIRESL  =$26
   HIRESH  =$27        STTEMP  =$FD        SVERT   =$FE        HORIZ   =$FF
   GETADR  =$4003      STARMOVE=$400E      STARMOVE1=$4028      STARMOVE2=$402F
   STAR    =$4030      STAR1   =$4032      START   =$404E      STARBYTE=$6000
   STARVERT=$6020      YVERTH  =$9200      YVERTL  =$92C0
```

LISTING #2

```
10  HGR
20  PRINT  CHR$ (4);"BLOAD STAR M
    OVE.OB"
25  PRINT  CHR$ (4);"BLOAD P1L"
30  FOR I = 24576 TO 24596:R =  INT
    ( RND (1) * 279): POKE I, INT
    (R / 7): NEXT
40  Y = 20
50  FOR I = 24608 TO 24628: POKE
    I,Y:Y = Y + 6: NEXT
60  CALL 16384
70  FOR J = 1 TO 20: NEXT
80  GOTO 60
```

LISTING #3

```
9200- 20 24 28 2C 30 34 38 3C
9208- 20 24 28 2C 30 34 38 3C
9210- 21 25 29 2D 31 35 39 3D
9218- 21 25 29 2D 31 35 39 3D
9220- 22 26 2A 2E 32 36 3A 3E
9228- 22 26 2A 2E 32 36 3A 3E
9230- 23 27 2B 2F 33 37 3B 3F
9238- 23 27 2B 2F 33 37 3B 3F
9240- 20 24 28 2C 30 34 38 3C
9248- 20 24 28 2C 30 34 38 3C
9250- 21 25 29 2D 31 35 39 3D
9258- 21 25 29 2D 31 35 39 3D
9260- 22 26 2A 2E 32 36 3A 3E
9268- 22 26 2A 2E 32 36 3A 3E
9270- 23 27 2B 2F 33 37 3B 3F
9278- 23 27 2B 2F 33 37 3B 3F
9280- 20 24 28 2C 30 34 38 3C
9288- 20 24 28 2C 30 34 38 3C
9290- 21 25 29 2D 31 35 39 3D
9298- 21 25 29 2D 31 35 39 3D
92A0- 22 26 2A 2E 32 36 3A 3E
92A8- 22 26 2A 2E 32 36 3A 3E
92B0- 23 27 2B 2F 33 37 3B 3F
92B8- 23 27 2B 2F 33 37 3B 3F

92C0- 00 00 00 00 00 00 00 00
92C8- 80 80 80 80 80 80 80 80
92D0- 00 00 00 00 00 00 00 00
92D8- 80 80 80 80 80 80 80 80
92E0- 00 00 00 00 00 00 00 00
92E8- 80 80 80 80 80 80 80 80
92F0- 00 00 00 00 00 00 00 00
92F8- 80 80 80 80 80 80 80 80
9300- 28 28 28 28 28 28 28 28
9308- A8 A8 A8 A8 A8 A8 A8 A8
9310- 28 28 28 28 28 28 28 28
9318- A8 A8 A8 A8 A8 A8 A8 A8
9320- 28 28 28 28 28 28 28 28
9328- A8 A8 A8 A8 A8 A8 A8 A8
9330- 28 28 28 28 28 28 28 28
9338- A8 A8 A8 A8 A8 A8 A8 A8
9340- 50 50 50 50 50 50 50 50
9348- D0 D0 D0 D0 D0 D0 D0 D0
9350- 50 50 50 50 50 50 50 50
9358- D0 D0 D0 D0 D0 D0 D0 D0
9360- 50 50 50 50 50 50 50 50
9368- D0 D0 D0 D0 D0 D0 D0 D0
9370- 50 50 50 50 50 50 50 50
9378- D0 D0 D0 D0 D0 D0 D0 D0
```

MUSIC CONSTRUCTION SET

背景音樂問題

麥法基

上期刊出有關背景音樂後，接到一些讀者說不能依著所說產生背景音樂，原因可能是程序上有所出入。筆者曾到高登商場知悉很多讀者的MCS MANNUAL 並不完全，所以在這裏更加詳細加以補充。

首先MCS 必須配備有MOCKINGBOARD 才能產生背景音樂。因爲MOCKINGBOARD 是獨立處理聲音，並不會干擾程式正常運行，除非程式內要利用I.O.（例如DOS COMMAND）操作。

現在請把你的MCS 放入磁碟機內BOOT 碟，當畫面上顯示MCS 的TITLE 時，按RETURN，鍵進入應用狀態。把手指的圖形指向磁碟圖形內，按RETURN 然後再鍵入CATALOG，MCS 的不同檔案便會顯現於螢幕上。讓我們看看CANON這首曲的檔案，該檔案分爲兩部份，雖兩個檔案都是BINARY FILE，上面一個是音樂符號圖形，以供MCS 演奏時顯示於畫面上。下面一個加上OBJ的是MOCKI—NGBOARD 資料，不過這個檔案只供MCS 自己所用，因爲除了資料外，還有其他聲量控制，音色和速度等資料在內，這一個亦不能給用者加進自己的程式裏。

跟著請鍵入LOAD CANON然後按RETURN讓MCS 把整首曲資料輸入，待輸入完畢後，要再按RETURN 返回MCS ，這時可把手指圖形指向鋼琴圖形讓整首曲演奏了。請耐心等待演奏完畢 （倘若中途按鍵，則會有一部份資料失去），把手指圖形指向磁碟圖形再按RETURN 進入應用狀態。這時需要把MCS 取出換上一張已經FORMATED的磁碟，再鍵入BSAVE CANON把自己需要的資料存入在你的磁碟內。請留意記着把MCS 取出，不要BSAVE到MCS內，否則會把MCS 內的CANON圖形資料刪去。

假如以上程序沒有錯漏，我們已經有一個可以用作背景音樂的CANON FILE 了。存在磁碟上的CANON只是資料而已，我們還需要一個產生音樂的程式，上一期已經把該程式的LISTING 列出，如果你想節省工夫，亦可以用FID的COPY PROGRAM把 MCS 上的 MUSIC 檔案抄錄在存有CANON資料的磁碟上加以運用。

在這裏筆者指出就是一些MCS 內可能沒有MUSIC 這個程式，原因是抄錄的MCS 並非在原庄碟抄出，倘若你發覺不能BSAVE 或找不到MUSIC 檔案，這是十分不幸的事，不過亦可到電腦時代查詢，看看可否帮你忙。

在未開始爲任何程式配上音樂時，請作下面一個試驗：按CALL—151進入監督程式，然後BLOAD MUSIC ，再BLOAD CANON，A$ 4000。跟着鍵入4900L，這時可以看到MUSIC的程式了。

鍵入4903G，CANON 這首曲立刻演奏起來了，不過總有些不像剛才MCS 所奏一樣，是嗎？聲音比較上有斷斷續續的感覺。因爲你BLOAD 的是一個純資料檔案，並未加上音色和速度，首先觀察4912的數值，這是04；請把它更改一下，鍵入4912：06（無須按RESET，MOCKINGBOARD會獨立處理音樂），整首曲忽然慢下來了，再把數值改爲01，效果如何？曲子忽然快得可笑！

現在大家知道4912是決定演奏的速度了，至於甚麼數值適合，大概你自己也可以找到吧。（04是最適合了）

試看4913的值，把它改爲20，曲子演奏像風琴的音色了，如果改爲01，則好像木琴的音色，換言之，4913是負責音色消逝的快慢。

再改改4918的數值，數值越大音量越大了，4918是音量的控制。

最後再更改491B 的數值，這個數值不能大過4918，因爲這是靜止時間。數值越少的話，音樂會變得斷續了。

最後一提是資料檔案 CANON 最先會放在$ 1000的位址，所以當BLOAD 時記着要加上A$ 4000。

音樂 HELLO

到目前爲止，我們可以著手爲自己的程式加上背景音樂了。這裏筆者借用兩個程式作爲例子，第一個最簡單莫如爲你的磁碟配上一個有音樂的HELLO ，就由你這隻磁碟開始吧。參看LISTING ＃1 ，這是一個幾乎所有使用APPLE的朋友都擁有的HELLO ，相信不用筆者解釋了。請在程式上加入第10，第20，第30，第135和136 行變成LISTING ＃2然後SAVE HELLO 到磁碟上。工作完畢後請重新BO—OT機，一首清脆的音樂陪伴閣下的HELLO 在演奏着，使整個電腦忽然間有生氣起來。

第10行是存入MUSIC和 CANON 並且輸入以上提過的音色，音量等資料。CALL 18691等於4903G，亦即呼叫音樂產生。

第30行的F＝0 讓音樂只在BOOT 機時演奏，因爲倘若閣下按1鍵時存入任何檔案後，可能會把$ 4000位址用去，這時無須再奏音樂了。

第136行的CALL 18700 是停止音樂，亦是監督程式的490CG ，這是必須的，因爲停止演奏後該程式會把MOC—KINGBOARD殘餘聲音消去。

如果讀者有興趣的話，更可以按鋼琴五線譜編造不同的樂曲配上你的HELLO上。

上面的例子顯然十分簡單，全個程式不超過20行，當然易如反掌了。如果配在比較大的程式內又如何呢？就拿上一期的SOUR GRAPE作爲另外一個例子吧。

通常遊戲配上背景音樂更加使人發生濃厚興趣。第一步當然是編造一首喜愛的樂曲了，樂曲最好盡量和遊戲配合例如快慢，節奏等。爲了簡單起見，就取用MCS 內的一首R—ACKET配上SOUR GRAPE上，請按談過的方法把RACK

背景音樂

一ＥＴ造一個資料檔案存放在你的磁碟上，就存在你的新的HELLO磁碟上，好嗎？

第二步要檢查一下遊戲的程式，最主要是程式是否要利用第三頁，通常是聲音程式。從第720至780行的SOUR GRAPE 我們知道該程式把聲音和圖形都放在＄300開始位址上，這樣假使配上背景音樂，會和MOCKINGBOARD資料有所衝突，所以程式必須作出適當修改。

請按LISTING ＃ 3的各行重新鍵入，第720至810 行把聲音發聲部份移去＄5000（20482）和圖形表移去＄6000（24576）位址。第８行和第９行輸入背景音樂和設定聲音資料，並且預先宣告PI是PITCH，TI是TIME，SOUND是APPLE喇叭程式。

其餘部份均與喇叭發聲有關，亦要跟着加以修改。最後把已經修改的程式沿用SOUR GRAPE存放上磁碟上，則大功告成。修改他人程式必須注意下列幾點：

㈠盡量把應該修改的地方更改，無須把整個程式重新編寫，亦是尊重他人作品的一個原則。

㈡留意與機械語溝通的地方，尤其是HIMEN和LOME—N，可能會因修改有所改變。有些程式會把APPLESOFT部份移往＄800以外的地址運行，如果發覺在＄4000開始，則須把它移往＄5000，否則會侵佔音樂部份。

㈢程式如果利用高解像第二頁，則無法更改。這時把筆者上一期的SOUREC CODE組合在 ＄6900或其他位址，而音樂資料例如這個RACKET便須要BLOAD 在＄7000位址內，解決記憶體問題。至於呼叫音樂部份當然再不是CA—LL 18691或CALL 18700 呼叫那一個位置上期有所提及，請參閱。（其他快慢，音色等亦不同了）

㈣如果發覺STRING字串，DIMENSION陣列等太多或太大，則最好利用RAM DOS把磁碟系統程式移上RAM CARD，這時HIMEN會改爲＄BF00開始，更適當的是把—LOMEN推上＄9600（38400），可以讓出大概10K以供字串和陣列之用，無須再担心記憶分佈問題。

總之修改必須留意每一個小節，最令人生氣的就是TY—PO ERROR，可能令程式不能運行而不明所以。

現在就試再爲其他程式配上音樂吧！那個CLOSING ALIEN 亦是一個很好的練習，試試看。

祝好運！ ■

LISTING ＃3

```
8  PRINT  CHR$ (4);"BLOAD MUSIC":
    PRINT  CHR$ (4);"BLOAD RACK
   ET,A$4000": POKE 18712,11: POKE
   18706,2: POKE 18707,3: POKE
   18715,1
9 PI = 20480:TI = 20481:SOUND = 2
   0482


340  IF CL% = 5 THEN  POKE PI,50:
    POKE TI,10: CALL SOUND:SC% =
   SC% + LE%:GE% = GE% + 1: GOSUB
   510: GOTO 290


350  IF CL% = 13 THEN  FOR L = 20
    TO 30: POKE PI,L: POKE TI,2
   : CALL SOUND: NEXT L:SC% = S
   C% + (10 * LE%):GE% = GE% +
   1:SE% = SE% + 1: GOSUB 510: GOTO
   290


460  FOR L = 1 TO 3: POKE PI,200:
    POKE TI,100: CALL SOUND: NEXT
   L: POKE PI,250: POKE TI,250:
    CALL SOUND


480  HTAB 1: VTAB 23: CALL 18700:
    PRINT "ANOTHER GAME(Y/N)? "
   ;: GET A$: IF A$ = "N" THEN
    TEXT : HOME : VTAB 22: END


500  CALL 18691:SC% = 0:LI% = 3:L
   E% = 1: GOTO 20


720  POKE 233,96: POKE 232,0


760  FOR L = 0 TO 23: READ L2: POKE
   20482 + L,L2: NEXT L
770  FOR L = 0 TO 191: READ L2: POKE
   24576 + L,L2: NEXT L
780 A$ = " SOUR GRAPE": FOR L = 1
    TO 240: POKE PI,L: POKE TI,
   7: CALL SOUND: PRINT A$;: NEXT
   L


810  DATA  174,0,80,173,48,192,13
   6,208,5,206,1,80,240,9,202,2
   08,245,174,0,80,76,5,80,96
```

背景音樂

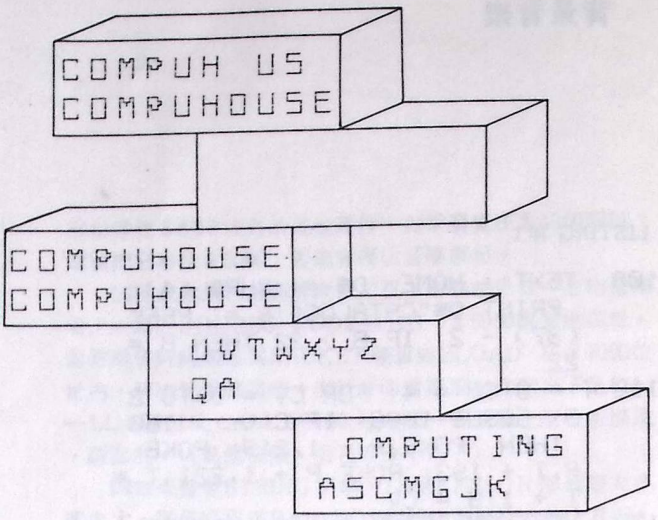
LISTING #1

```
100  TEXT : HOME :D$ =  CHR$ (4):
      PRINT D$"CATALOG":B =  PEEK
     (37) - 2: IF B > 22 THEN B =
     22
110 T = 0:CH = 4: FOR CV = 0 TO 2
     3: GOSUB 1000: IF C <  > 160
      THEN  POKE P - 1,219: POKE
     P,T + 193: POKE P + 1,221:T =
     T + 1:S = CV
120  NEXT CV: VTAB 24:A$ = "TYPE
     LETTER TO RUN  TYPE LETTER T
     O RUN 1=LOAD 2=LOCK 3=UNLOCK
      4=DELETE 5=EXIT "
130 B$ = "RUN": HTAB 1: PRINT  LEFT$
     (A$,39);:A$ =  MID$ (A$,2) +
      LEFT$ (A$,1):K =  PEEK ( -
     16384): IF K < 128 THEN  FOR
     K = 1 TO 75: NEXT K:K =  FRE
     (0): GOTO 130
140  POKE  - 16368,0:K = K - 176:
      IF K < 1 OR K > 5 THEN 300
200  HTAB 1: CALL  - 868: IF K =
     5 THEN  END
210  PRINT "PRESS 'LETTER' YOU WI
     SH TO ";: IF K = 1 THEN B$ =
     "LOAD"
220  IF K = 2 THEN B$ = "LOCK"
230  IF K = 3 THEN B$ = "UNLOCK"
240  IF K = 4 THEN B$ = "DELETE":
      FLASH
250  PRINT B$;: CALL  - 198: NORMAL
     : GET K$:K =  ASC (K$) - 48
300  IF K < 17 OR K > T + 16 THEN
     130
310 CH = 1:CV = S - T + K - 16: GOSUB
     1000: IF C = 194 AND (B$ = "
     RUN" OR B$ = "LOAD") THEN B$
      = "B" + B$
320  FOR CH = 6 TO 39: GOSUB 1000
     :B$ = B$ +  CHR$ (C): NEXT C
     H: HTAB 1: CALL  - 868: PRINT
     B$: PRINT D$;B$: GOTO 100
1000 C1 =  INT (CV / 8):C2 = CV -
     C1 * 8:P = 1024 + 128 * C2 +
     40 * C1 + CH:C =  PEEK (P): RETURN
```

LISTING #2

```
10  HOME : PRINT  CHR$ (4);"BLOAD
     MUSIC": PRINT  CHR$ (4);"BL
     OAD CANNON,A$4000": POKE 187
     07,48: POKE 18712,9
20  PRINT  CHR$ (4);"CATALOG":B =
     PEEK (37) - 2: CALL 18691:B
     =  PEEK (37) - 2: IF B > 22
     THEN B = 22
30 F = 0: GOTO 110
100  TEXT : HOME :D$ =  CHR$ (4):
      PRINT D$"CATALOG":B =  PEEK
     (37) - 2: IF B > 22 THEN B =
     22
110 T = 0:CH = 4: FOR CV = 0 TO 2
     3: GOSUB 1000: IF C <  > 160
      THEN  POKE P - 1,219: POKE
     P,T + 193: POKE P + 1,221:T =
     T + 1:S = CV
120  NEXT CV: VTAB 24:A$ = "TYPE
     LETTER TO RUN  TYPE LETTER T
     O RUN 1=LOAD 2=LOCK 3=UNLOCK
      4=DELETE 5=EXIT "
130 B$ = "RUN": HTAB 1: PRINT  LEFT$
     (A$,39);:A$ =  MID$ (A$,2) +
      LEFT$ (A$,1):K =  PEEK ( -
     16384): IF K < 128 THEN  FOR
     K = 1 TO 75: NEXT K:K =  FRE
     (0): GOTO 130
135  IF F > 0 THEN 140
136  CALL 18700:F = 1
140  POKE  - 16368,0:K = K - 176:
      IF K < 1 OR K > 5 THEN 300
200  HTAB 1: CALL  - 868: IF K =
     5 THEN  END
210  PRINT "PRESS 'LETTER' YOU WI
     SH TO ";: IF K = 1 THEN B$ =
     "LOAD"
220  IF K = 2 THEN B$ = "LOCK"
230  IF K = 3 THEN B$ = "UNLOCK"
240  IF K = 4 THEN B$ = "DELETE":
      FLASH
250  PRINT B$;: CALL  - 198: NORMAL
     : GET K$:K =  ASC (K$) - 48
300  IF K < 17 OR K > T + 16 THEN
     130
310 CH = 1:CV = S - T + K - 16: GOSUB
     1000: IF C = 194 AND (B$ = "
     RUN" OR B$ = "LOAD") THEN B$
      = "B" + B$
320  FOR CH = 6 TO 39: GOSUB 1000
     :B$ = B$ +  CHR$ (C): NEXT C
     H: HTAB 1: CALL  - 868: PRINT
     B$: PRINT D$;B$: GOTO 100
1000 C1 =  INT (CV / 8):C2 = CV -
     C1 * 8:P = 1024 + 128 * C2 +
     40 * C1 + CH:C =  PEEK (P): RETURN
```

兒童串字訓練



兒童在日常溫習中都耗費不少時間來學習英文生字，爲了節省學習的時間，提高學習的興趣，筆者特別編寫出一個程式作輔導兒童來學習生字之用。

本程式會隨意選擇它其中的生字表並將這個生字顯示到螢幕上予兒童學習，之後電腦會要求兒童將這個生字鍵入，若果這個字串鍵入正確的話，則電腦會選擇另一個生字。若果其中有些字串錯，電腦會在該錯誤字串的位置上出現"—"符號，例如兒童將ELEPHANT誤作爲ELAPHANT，則電腦會出現—EL—PHANT，然後會要求兒童再次將它鍵入。這個過程會持續直至輸入完全正確爲止。此時電腦會再次出現這個生字，不過電腦以隨機方式將其中一個字母略去，要求兒童再次鍵入它，如正確鍵入則電腦會再出現該生字，但今次會多漏空一個字母，這個過程會重覆直到兒童能記憶到這個生字爲止。筆者相信這個過程會對兒童學習一些較艱深的生字十分有效。

本程式在編寫的時候考慮到若果在每一次學習之中太多生字的話，是會令兒童感到煩擾的，因此程式只會發問十個生字，完成後電腦會計算成績結果。另一方面，本程式是採用高解像螢幕來顯示生字，字字建立在圖形表中，這樣可以增加美觀感覺，令兒童的趣味性提高。

執行本程式只需鍵入RUN SPEL—LING，輸入十個要兒童學習的生字後便可給兒童進行練習了。

本程式，共有兩個程式列表，一個是APPLESOFT 主程，一個是機械碼圖形表。

前者，大家於打好後可用下列指令存入磁碟：

SAVE SPELLING。

至於後者（圖形表），大家可用下列指令存入磁碟：

BSAVE ALPHABET，A$1000，L$281

有關程式列表的輸入方法可參考本期的〔輸入程式方法〕。■

```
******************
**    ALPHABET    **
**                **
******************
1000- 20 41 4C 4C 4F 57 45 44
1008- 20 49 4E 20 54 48 49 53
1010- 20 4C 49 53 54 2E 22 3A
1018- BA 3A BA 22 59 4F 55 20
1020- 48 41 56 45 20 32 30 2E
1028- 22 3A AB 36 38 30 00 5A
1030- 10 44 02 97 3A A2 35 3A
1038- BA 22 52 45 4D 45 4D 42
1040- 45 52 20 54 4F 20 54 59
1048- 50 45 20 54 48 45 20 57
1050- 4F 52 44 27 44 4F 4E 45
1058- 27 00 8B 10 4E 02 BA 3A
1060- BA 22 41 53 20 54 48 45
1068- 20 4C 41 53 54 20 57 4F
1070- 52 44 20 49 4E 20 59 4F
1078- 55 52 20 53 50 45 4C 4C
1080- 49 4E 47 20 4C 49 53 54
1088- 2E 22 00 93 10 58 02 BA
1090- 3A BA 00 C9 10 62 02 BA
1098- 3A BA 22 54 59 50 45 20
10A0- 49 4E 20 57 4F 52 44 20
10A8- 22 3B 4E 3B 22 2E 22 3A
10B0- BA 3A 84 22 54 48 45 20
10B8- 57 4F 52 44 20 49 53 3F
10C0- 20 22 3B 57 24 28 4E 29
10C8- 00 CF 10 6C 02 BA 00 21
10D0- 11 76 02 AD E3 28 57 24
10D8- 28 4E 29 29 CF 31 30 C4
10E0- BA 22 4D 41 58 49 4D 55
10E8- 4D 20 57 4F 52 44 20 4C
10F0- 45 4E 47 54 48 20 49 53
10F8- 20 54 45 4E 20 4C 45 54
1100- 54 45 52 53 2E 22 3A BA
1108- 3A BA 22 43 48 4F 4F 53
1110- 45 20 41 4E 4F 54 48 45
1118- 52 2E 22 3A AB 36 31 30
1120- 00 75 11 80 02 BA 3A AD
1128- E3 28 57 24 28 4E 29 29
1130- D1 D0 31 C4 BA 22 57 4F
1138- 52 44 53 20 4D 55 53 54
1140- 20 48 41 56 45 20 41 54
1148- 20 4C 45 41 53 54 20 22
1150- 3A BA 3A BA 22 54 57 4F
1158- 20 4C 45 54 54 45 52 53
1160- 2E 20 54 52 59 20 41 4E
1168- 4F 54 48 45 52 2E 22 3A
1170- AB 36 31 30 00 8B 11 8A
1178- 02 AD 57 24 28 4E 29 D0
1180- 22 44 4F 4E 45 22 C4 36
1188- 38 30 00 94 11 94 02 AB
1190- 35 36 30 00 9A 11 9E 02
1198- BA 00 BE 11 A8 02 BA 3A
11A0- BA 22 4D 49 53 54 41 4B
11A8- 45 53 3F 20 54 59 50 45
11B0- 20 57 4F 52 44 53 20 4F
11B8- 56 45 52 3F 22 00 D3 11
11C0- B2 02 BA 3A 84 22 28 59
11C8- 20 4F 52 20 4E 29 22 3B
11D0- 5A 24 00 EB 11 BC 02 AD
11D8- E8 28 5A 24 2C 31 29 D0
11E0- 22 59 22 C4 BD 3A AB 35
11E8- 35 30 00 00 12 C6 02 AD
11F0- E8 28 5A 24 2C 31 29 D0
11F8- 22 4E 22 C4 37 33 30 00
1200- 09 12 D0 02 AB 36 38 30
1208- 00 30 12 DA 02 97 3A A2
1210- 35 3A BA 22 53 54 55 44
1218- 45 4E 54 53 20 43 41 4E
1220- 20 4E 4F 57 20 50 52 41
1228- 43 54 49 43 45 20 22 00
1230- 38 12 E4 02 52 D0 30 00
1238- 55 12 EE 02 BA 3A BA 22
1240- 53 50 45 4C 4C 49 4E 47
1248- 20 54 48 45 20 57 4F 52
1250- 44 53 2E 22 00 81 12 F8
1258- 02 A2 32 31 3A 9F 3A 84
1260- 22 50 52 45 53 53 20 52
1268- 45 54 55 52 4E 20 57 48
1270- 45 4E 20 52 45 41 44 59
1278- 22 3B 5A 24 3A 9D 3A 97
1280- 00 8C
```

兒童串字訓練

```
1  REM  ************************
2  REM  *                      *
3  REM  *       SPELLING       *
4  REM  *                      *
5  REM  *  COPYRIGHT 1985 (C)  *
6  REM  *  BY COMPUTING AGE    *
7  REM  *                      *
8  REM  ************************

100  REM
110  CLEAR : TEXT
120 D$ =  CHR$ (4)
130  PRINT D$;"BLOAD ALPHABET,A$8
     4DO"
140  POKE 115,208: POKE 116,132
150  POKE 232,208: POKE 233,132
160  REM  W$ IS THE WORD RANDOMLY
      CHOSEN FROM WORD LIST.
170  REM  W$(21) IS ARRAY OF SPEL
     LING WORDS TYPED IN BY TEACH
     ER.
180  REM  W1$=THE WORD BEING SPEL
     LING WITH MORE AND MORE LETT
     ERS REMOVED FROM IT.
190  REM  W2$=TEMPORARY STORAGE O
     F WORD CHOSEN EACH TIME FOR
     RETRIEVAL.
200  REM  Z$=CARRIAGE RETURN OR A
     NY KEY PRESS BY OPERATOR.
220  REM  CH=RND NO. OF CHARACTER
      IN A WORD TO BE REMOVED AS
     1ST, 3RD, ETC.
230  REM  CN=LETTER OF WORD TO BE
      DRAWN IN HIRES.
240  REM  DO IS REDRAWING OF WORD
      ONE TIME FOR EACH LETTER WI
     TH ONE LETTER LESS EACH TIME
      UNTIL SCREEN BLANK
250  REM  LE=LENGTH OF LONGEST WO
     RD WHETHER FROM SPELLING LIS
     T OR STUDENT INPUT.
260  REM  LE ENABLES DASHES IF ST
     UDENT ANSWER IS LONGER OR SH
     ORTER THAN RIGHT ANSWER.
270  REM  X=FRE(0) UUSED TO REMOV
     E STRING ACCUMULATION FROM S
     TUDENT ANSWERS.
280  REM  SE(N) AND SP(N) ARE SWI
     TCHES
290  REM  XN AND YN ARE COORDINAT
     ES OF EACH LETTER DRAWN
300  REM  R IS STUDENT ERROR COUN
     T.
310  REM  WN IS RANDOM NO. GENERA
     TED TO SELECT SPELLING WORD.
320  REM  X=FRE(0) GETS RID OF AC
     CUMULATING STRINGS TO AVOID
     MEMORY FULL ERROR.
330  LOMEM: 17000
340  PRINT
350  REM  LOMEM SET ABOVE GRAPHIC
     S AREA TO AVOID INTER FERENC
     E WITH HGR
360 X =  FRE (0): HOME
370  DIM W$(21): DIM SP(21)
380  VTAB 23: PRINT "** COPYRIGHT
       1984 BY COMPUTING AGE  **"
     : VTAB 2: HTAB 15: PRINT "DI
     RECTIONS": PRINT
390  PRINT "DEAR TEACHER:": PRINT

400  PRINT "THIS PROGRAM ENABLES
     U TO TYPE IN": PRINT
410  PRINT "UP TO 20 SPELLING WOR
     D FOR THE DAY'S": PRINT
420  PRINT "SPELLING PRACTICE."
430  PRINT
440  PRINT : PRINT "THE LAST WORD
      YOU TYPE IN SHOULD BE": PRINT

450  PRINT "THE WORD 'DONE'."
460  PRINT : PRINT "EACH LESSON I
     S TEN WORD."
470  VTAB 20: FLASH : INPUT "PRES
     S RETURN TO GO ON.";Z$: NORMAL

480  HOME
490  VTAB 5
500  PRINT "READY TO TYPE IN SPEL
     LING LIST?
510  PRINT : INPUT "(Y OR N) ";Z$

520  IF  LEFT$ (Z$,1) = "Y" THEN
     550
530  IF  LEFT$ (Z$,1) = "N" THEN
      PRINT : PRINT "THAT'S ALL."
     : END
540  GOTO 510
550 N = 0
560 N = N + 1
570  IF N = 21 THEN  PRINT "NO MO
     RE WORDS ALLOWED IN THIS LIS
     T.": PRINT : PRINT "YOU HAVE
      20.": GOTO 680
580  HOME : VTAB 5: PRINT "REMEMB
     ER TO TYPE THE WORD'DONE'
590  PRINT : PRINT "AS THE LAST W
     ORD IN YOUR SPELLING LIST."
600  PRINT : PRINT
610  PRINT : PRINT "TYPE IN WORD
     ";N;".": PRINT : INPUT "THE
     WORD IS? ";W$(N)
620  PRINT
630  IF  LEN (W$(N)) > 10 THEN  PRINT
     "MAXIMUM WORD LENGTH IS TEN
     LETTERS.": PRINT : PRINT "CH
     OOSE ANOTHER.": GOTO 610
```

兒童串字訓練

```
640  PRINT : IF  LEN (W$(N)) <  =
     1 THEN  PRINT "WORDS MUST HA
     VE AT LEAST ": PRINT : PRINT
     "TWO LETTERS. TRY ANOTHER.":
      GOTO 610
650  IF W$(N) = "DONE" THEN 680
660  GOTO 560
670  PRINT
680  PRINT : PRINT "MISTAKES? TYP
     E WORDS OVER?"
690  PRINT : INPUT "(Y OR N)";Z$
700  IF  LEFT$ (Z$,1) = "Y" THEN
      CLEAR : GOTO 550
710  IF  LEFT$ (Z$,1) = "N" THEN
     730
720  GOTO 680
730  HOME : VTAB 5: PRINT "STUDEN
     TS CAN NOW PRACTICE "
740 R = 0
750  PRINT : PRINT "SPELLING THE
     WORDS."
760  VTAB 21: FLASH : INPUT "PRES
     S RETURN WHEN READY";Z$: NORMAL
     : HOME
770 NW = N - 1
775  REM  NEXT STATEMENT USED AS
     SWITCH
780  FOR J = 1 TO NW:SP(J) = 1: NEXT
     J
790  IF NW < 10 THEN  FOR I = 1 TO
     NW: GOTO 810
800  FOR I = 1 TO 10
810 WN =  INT (NW *  RND (1)) + 1
830  IF SP(WN) = 0 THEN 810
840 SP(WN) = 0
850 W$ = W$(WN)
860 W2$ = W$(WN)
870  HGR : HCOLOR= 3: ROT= 0: SCALE=
     3
880 XN = 10:YN = 30
890 LT =  LEN (W$)
900  REM  THIS DESIGNED TO AVOID
     ELIMINATING SAME LETTER.
910  FOR J = 1 TO LT:SE(J) = 1: NEXT
     J
920  REM  NEXT STATEMENT IS TO EV
     ENTUALLY PUT IN AS MANY BLAN
     KS AS THE WORD HAS LETTERS.
930  FOR DO = 1 TO LT
940  FOR CN = 1 TO LT
950  HOME
960  DRAW  ASC ( MID$ (W$,CN,1)) -
     31 AT XN,YN
970 XN = XN + 25
980  NEXT CN
990 XN = 10:YN = 60
1000  VTAB 21: INPUT "TYPE THE WO
     RD.";TW$
1010  VTAB 22: IF  LEN (TW$) > 10
      THEN  PRINT "YOUR WORD IS T
     OO LONG. TRY AGAIN.": GOTO 1
     000
1020  VTAB 22: IF  LEN (TW$) <  =
     1 THEN  PRINT "YOUR WORD IS
     TOO SHORT. TRY AGAIN.": GOTO
     1000
1030  FOR CN = 1 TO  LEN (TW$)
1040  DRAW  ASC ( MID$ (TW$,CN,1)
     ) - 31 AT XN,YN
1050 XN = XN + 25
1060  IF XN > 260 THEN  PRINT "WO
     RD IS TOO LONG. TRY AGAIN.":
      GOTO 780
1070  NEXT CN
1080  IF TW$ = W2$ THEN 1240
1090 R = R + 1
1100  REM  START COMPARING EACH
     LETTER IN STUDENT ANSWER WIT
     H RIGHT ANSWER 'W2$'.
1110 XN = 10:YN = 90
1120  IF  LEN (W$) >  =  LEN (TW$
     ) THEN LE =  LEN (W$): GOTO
     1140
1130 LE =  LEN (TW$)
1140  FOR CN = 1 TO LE
1150  IF  MID$ (TW$,CN,1) =  MID$
     (W2$,CN,1) THEN  DRAW  ASC (
      MID$ (TW$,CN,1)) - 31 AT XN
     ,YN: GOTO 1170
1160  DRAW  ASC ("-") - 31 AT XN,
     YN
1170 XN = XN + 25
1180  NEXT CN
1190  VTAB 22
1200  PRINT "WRONG! SEE THE MISSI
     NG LETTERS.": PRINT : PRINT

1210  PRINT "EACH DASH IS A MISTA
     KE.": PRINT
1220  INPUT "RETURN TO GO ON.";Z$

1230  CALL 62450:XN = 10:YN = 30:
     GOTO 940
1240  REM  WE GO ON FROM HERE TO
     REMOVE BLANKS
1250  PRINT
1260  VTAB 22: PRINT "RIGHT!": INPUT
     "PRESS RETURN TO GO ON.";Z$
1270  PRINT
1275  REM  CH IS RANDOMLY SELECTE
     D LETTER OR CHARACTER TO BE
     REMOVED FROM THE WORD.
1280 CH =  INT (LT *  RND (1) + 1
     )
1290  REM  AVOIDS REPEATING REMOV
     AL OF A BLANK ALREADY IN A W
     ORD
1300  IF SE(CH) <  > 1 THEN 1280
1310 SE(CH) = 0
1320 X$ = " "
1325  REM  NEXT STATEMENT INSERTS
      BLANK
```

兒童串字訓練

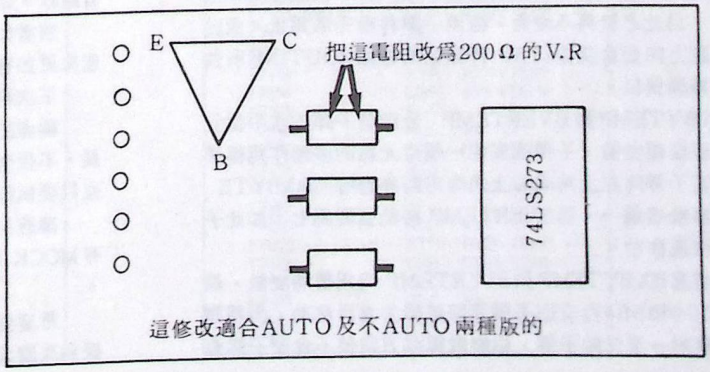
```
1330  IF CH = 1 THEN W$ = X$ +  RIGHT$
     (W$,LT - 1): GOTO 1350
1340 W$ =  LEFT$ (W$,CH - 1) + X$
      +  MID$ (W$,CH + 1)
1350  PRINT
1360 XN = 10:YN = 30
1370  CALL 62450: NEXT DO
1380  PRINT
1390  PRINT "NOW TYPE THE COMPLET
     E WORD FROM MEMORY.": INPUT
     TW$
1400  PRINT : PRINT
1410  IF TW$ = W2$ THEN  PRINT "Y
     OU NOW KNOW HOW TO SPELL '";
     TW$;"'.": GOTO 1470
1420  PRINT "SORRY. DID YOU CONCE
     NTRATE?"
1430  INPUT "RETURN TO GO ON.";Z$

1440  PRINT : PRINT
1450 R = R + 1: VTAB 24: PRINT "T
     HE WORD WAS '";W2$;"'."
1460  PRINT : INPUT "PRESS RETURN
      TO TRY IT AGAIN.";Z$: GOTO
     1390
1470  INPUT "RETURN TO GO ON.";Z$
     : NEXT I
1480  PRINT
1490  IF NW < 10 THEN  PRINT "YOU
      MADE ";R;" ERRORS ON THE ";
     NW" WORDS.": GOTO 1510
1500  PRINT "YOU MADE ";R;" ERROR
     S ON THE TEN WORDS."
1510  PRINT
1520  INPUT "MOVE PRACTICE? (Y OR
      N)";Z$
1530  IF  LEFT$ (Z$,1) = "Y" THEN
      TEXT : HOME :X =  FRE (0): GOTO
     730
1540  IF  LEFT$ (Z$,1) <  > "N" THEN
     1520
1550  PRINT
1560  VTAB 22: PRINT  FRE (0)
1570  HOME : TEXT : PRINT "THAT'S
      ALL!": END
1580  PRINT : PRINT : INPUT "RETU
     RN TO GO ON.";Z$
1590  PRINT "YOU MADE ";R;" ERROR
     S ON THE ";NW";"W OR DS." ■
```

調節80字行咭的光暗

沈耀峯

編者註：各位讀者目前在坊間買到的80字行咭是模仿外國一隻牌子。而原裝貨本來就是採用可變電阻(V.R)的，但本港的仿製品爲了減低成本，才用一枚固定電阻。

相信各位都遇過，80COLCARD的光度與電腦的光度有所不同，如過光或過暗。雖然你可以調節監視器的光度，但適合80COL 的光度時，又不適合40COL 的光度。又或者你可以調節電腦內的輸出V.R，但輸出有限，如調節後仍然覺80COL 過光，則你可以如圖㈠所示，把咭上的電阻改爲V.R，則你便可以調節這V.R 直至你滿意爲止，而不影响其它部份。■

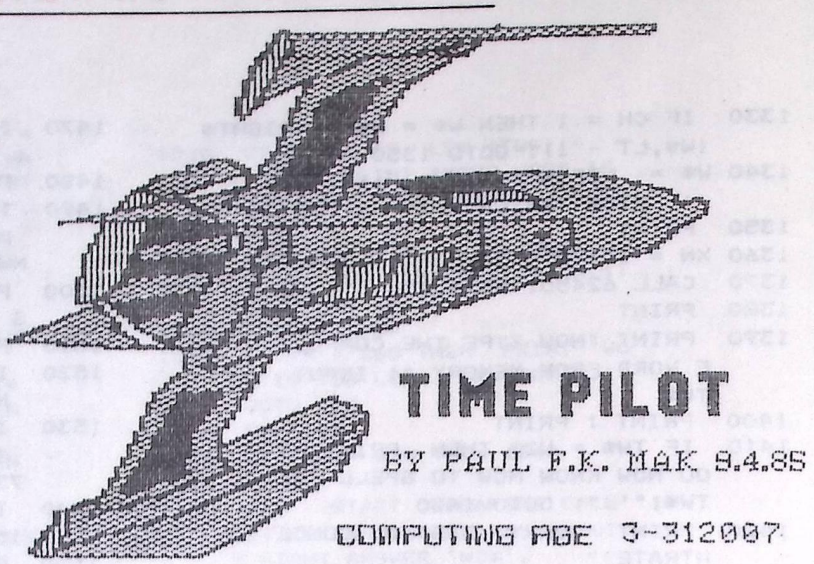


這修改適合AUTO及不AUTO兩種版的

GRAPHICS MAGICIAN編寫的遊戲：

TIME PILOT

麥法基



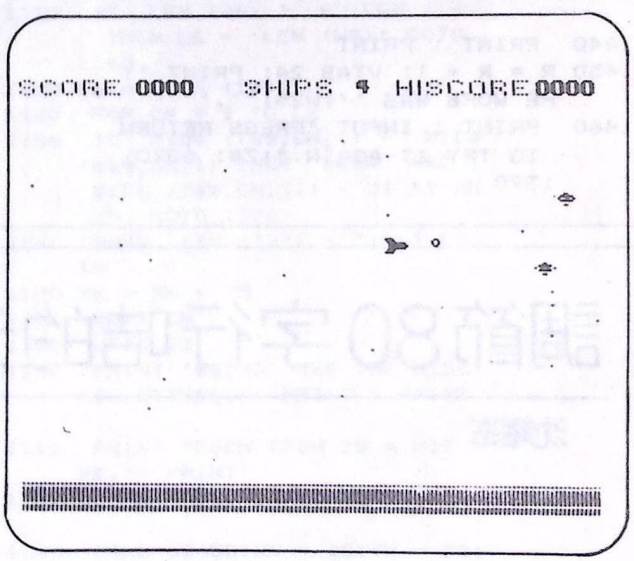
TIME PILOT 是一個緊張刺激的射擊遊戲，玩法無須在此描述，只要BOOT 起遊戲則小孩子也知道如何進行。不過最主要是把敵人母船消滅才能進入較高難度而已。

這個遊戲特色是你控制的戰機是可以四方八面發射子彈，如此可增加遊戲趣味。在編寫過程當中，受控制戰機的圖形有八個不同方向，換言之，子彈發射方向由圖形決定。請參看LISTING的PBULLETI 程式，這是負責子彈的發射過程。當第499行測試BUTTON後，程式跳到504行的PFIRE發射子彈，這時並不需要理會子彈方向，因爲是第一次繪畫出子彈圖形。子彈圖形畫出畫面後，第511行的PB—FLAG1 便會由零變爲大於零，表示子彈已經存在於螢幕上了。於是當程式再次運行時，第497行便會因爲旗號大於零的原故跳到第531行的PIMOVE 去移動子彈，這時便需要立即判斷子彈方向。子彈方向由548行的BDIRETION所負責，我們看看第1708 行的DIFLAG，該旗號的值由512行的PSHAPE所供應。PSHAPE是戰機方向圖形號碼，爲甚麼要另外設立一個DIFLAG來控制子彈方向而不直接使用戰機圖形呢？這不是多出一個旗號嗎？

因爲戰機方向每秒鐘都不會相同，而子彈方向一經發射之後便會跟着一定的方向移動，倘若直接使用戰機圖形來負責子彈移動方向的話，則子彈會跟隨戰機方向變動，這是不合實際環境。所以子彈一經發射後，DIFLAG便立即獲取戰機方向，自此之後再不變更，直至子彈移動完畢爲止。至於子彈移動方向便會交由1732 行開始的各個ROUTINE所負責給予繪圖資料。

BXBYTEMP和BVERTEMP 分別是子彈的水平位元組和垂直座標變數，子彈速度由一個位元組的移動作爲標準，所以當子彈向左上角或右上角等方向移動時，BXBYTE—MP 移動值爲一，而BVERTEMP 移動值則爲七，如此子彈才會快速移動。

請留意BXBYTEMP和BVERTEMP均爲暫時變數，最後會由599和564行交回子彈實際座標去進行移動，因爲戰機可以發射一至三粒子彈，由變數暫存方向值不會使子彈有所混亂。



以上解釋可能使某些讀者難於了解，不過筆者每一個遊戲實在太長了，相信假如要整個程式作出詳細註解，勢必要數拾頁紙之多。所以只能把其中特別的地方加以指出，希望讀者原諒。如要程式的原始碼，請到電腦時代詢問。

讀者如果有MOCKINGBOARD在第4 SLOT上，則玩起來更加有現場感受。

下次再會！

編者註：本遊戲由一組共9個FILE 組成，由於程式很長，不便全部刊出，大家可在今期程式磁碟找到本遊戲，大家只要執行TIME PILOT HELLO便可。

擁有MOCKINGBOARD的讀者敬請留意，這個遊戲是有MOCKING BOARD 版本的，大家可在遊戲開始時選擇。

希望要本程式的原始碼檔案的讀者，可携同今期程式磁碟到來讀者服務部購買，每張連磁碟十五元。■

TIME PILOT

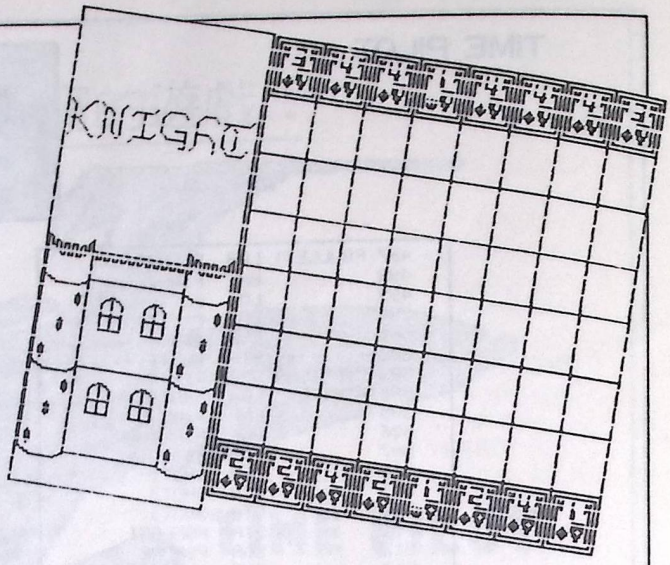
〔代郵〕
熱愛學習電腦遊戲程式寫作的人：
麥先生所編寫各個遊戲的SOURCE CODE 可到讀者服務部購買，每個十五元（連磁碟），購買時必須要携同該期原版磁碟，我們才會出售。

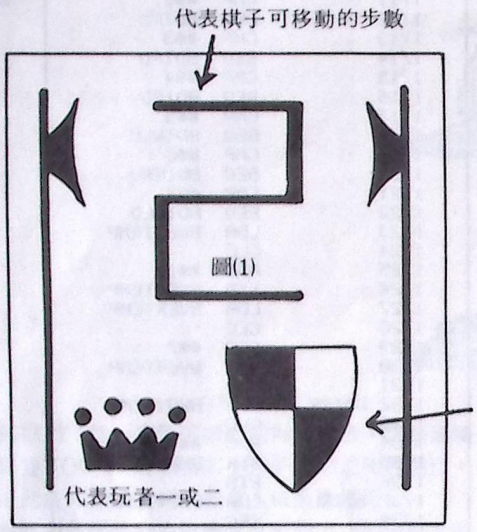
```
497 PBULLET1 LDA  PBFLAG1
498          BNE  P1MOVE
499          LDA  BUTTON
500          BMI  P1FIRE
501          LDA  #$00
502          STA  SHOT1
503 P1RETURN JMP  P1STAY
504 P1FIRE   LDA  SHOT1
505          CMP  #$01
506          BGE  P1RETURN
507          LDA  #$01
508          STA  SHOT1
509          STA  SHOT2
510          STA  SHOT3
511          STA  PBFLAG1
512          LDA  PSHAPE
513          STA  DIFLAG1
514          LDA  PLANEBYTE
515          CLC
516          ADC  #$1
517          STA  P1BYTE
518          STA  XBYTE
519          LDA  #$0
520          STA  XBIT
521          LDA  PLANEVERT
522          CLC
523          ADC  #$6
524          STA  P1VERT
525          STA  VERT
526          LDA  #$D
527          STA  SHNUM
528          JSR  $8F00
529          JSR  BURST
530          JMP  NOSHOT1
531 P1MOVE   LDA  C1FLAG
532          BNE  P1STAY
533          LDA  P1BYTE
534          STA  XBYTE
535          LDA  #$00
536          STA  XBIT
537          LDA  P1VERT
538          STA  VERT
539          LDA  #$10
540          STA  SHNUM
541          JSR  $8F00
542          LDA  DIFLAG1
543          STA  DIFLAG
544          LDA  P1BYTE
545          STA  BXBYTEMP
546          LDA  P1VERT
547          STA  BVERTEMP
548          JSR  BDIRECTION
549          LDA  BXBYTEMP
550          CMP  #$26
551          BGE  P1STAY
552          CMP  #$2
553          BLT  P1STAY
554          LDA  BVERTEMP
555          CMP  #$AF
556          BGE  P1STAY
557          CMP  #$C
558          BLT  P1STAY
559          LDA  BXBYTEMP
560          STA  P1BYTE
561          STA  XBYTE
562          LDA  #$0
563          STA  XBIT
564          LDA  BVERTEMP
565          STA  P1VERT
566          STA  VERT
567          LDA  #$D
568          STA  SHNUM
569          JSR  $8F00
570          JSR  FIND1
571          JMP  NOSHOT1
572 P1STAY   LDA  #$00
573          STA  PBFLAG1
574          LDA  #$80
575          STA  P1BYTE
576 NOSHOT1  RTS
```

```
1708 BDIRECTION LDA DIFLAG
1709            CMP  #$1
1710            BEQ  BDIRR
1711            CMP  #$2
1712            BEQ  BDIRL
1713            CMP  #$3
1714            BEQ  BDIRU
1715            CMP  #$4
1716            BEQ  BDIRD
1717            CMP  #$5
1718            BEQ  BDIRLU
1719            CMP  #$6
1720            BEQ  BDIRRU
1721            CMP  #$7
1722            BEQ  BDIRLD
1723            LDA  BXBYTEMP
1724            CLC
1725            ADC  #$1
1726            STA  BXBYTEMP
1727            LDA  BVERTEMP
1728            CLC
1729            ADC  #$7
1730            STA  BVERTEMP
1731            RTS
1732 BDIRR      LDA  BXBYTEMP
1733            CLC
1734            ADC  #$1
1735            STA  BXBYTEMP
1736            RTS
1737 BDIRL      LDA  BXBYTEMP
1738            SEC
1739            SBC  #$1
1740            STA  BXBYTEMP
1741            RTS
1742 BDIRU      LDA  BVERTEMP
1743            SEC
1744            SBC  #$7
1745            STA  BVERTEMP
1746            RTS
1747 BDIRD      LDA  BVERTEMP
1748            CLC
1749            ADC  #$7
1750            STA  BVERTEMP
1751            RTS
1752 BDIRLU     LDA  BXBYTEMP
1753            SEC
1754            SBC  #$1
1755            STA  BXBYTEMP
1756            LDA  BVERTEMP
1757            SEC
1758            SBC  #$7
1759            STA  BVERTEMP
1760            RTS
1761 BDIRRU     LDA  BXBYTEMP
1762            CLC
1763            ADC  #$1
1764            STA  BXBYTEMP
1765            LDA  BVERTEMP
1766            SEC
1767            SBC  #$7
1768            STA  BVERTEMP
1769            RTS
1770 BDIRLD     LDA  BXBYTEMP
1771            SEC
1772            SBC  #$1
1773            STA  BXBYTEMP
1774            LDA  BVERTEMP
1775            CLC
1776            ADC  #$7
1777            STA  BVERTEMP
1778            RTS
1779 BDIRRD     LDA  BXBYTEMP
1780            CLC
1781            ADC  #$1
1782            STA  BXBYTEMP
1783            LDA  BVERTEMP
1784            CLC
1785            ADC  #$7
1786            STA  BVERTEMP
1787            RTS
```

武士忠魂

陳文緯





這個遊戲不屬刺激性，主要是考考大家的智慧，可供兩人玩的。

開始時，電腦會繪出一個堡壘，跟着奏出一首動聽THE END OF WOR—LD 音樂，然後繪出一個8×8的棋盤，每邊放八隻棋子。

所有棋子繪畫後，便出現一個閃動的CURSOR，PLAYER 1 可用IJKM四鍵選擇要行那一隻棋子，RETURN決定，棋中右下角的盾形，是用來代表那個PLAYER ；左下邊一個符號代表他的身份，身份分皇帝和武士，此外上半部是一個由1—4的數目。大家可從圖1中看到。

選擇完要行的棋後，再以IJKM 鍵來控制這棋，行幾步是照棋上的數目，棋子走動完後每次都會隨機更改數目的。ESC 鍵是放棄行這隻棋，而重新選擇另一隻棋。

每次行棋的路綫不能重覆（即回頭），而且一定要行指令數目的步數，如果最後一步可以的話，可以吃了另一個PLAYER 的棋，吃到皇帝便勝出。否則，便要把行棋權交到另一PLAYER手中。

本程式是以APPLESOFT 編寫，另外一個SHAPE TABLE ，主要劃棋子、堡壘和KNIGHT 字。FILENAME名稱是KNIGHT和KNIGHT SHAPES。讀者在SAVE KNIGHT SHAPES入磁碟時，可用下列指令

BSAVE KNIGHT SHAPES，A$6000，L$01E5。■

```
******************
*  KNIGHT SHAPE *
*  A$6000,L$1E5 *
******************
6000- 15 00 2C 00 34 00 42 00
6008- 50 00 5D 00 C4 01 7D 00
6010- 82 00 8C 00 B0 00 BE 00
6018- CB 00 D9 00 E5 00 F7 00
6020- 0A 01 29 01 46 01 62 01
6028- 83 01 9E 01 09 25 27 25
6030- 27 24 04 00 2D 2D E5 DB
6038- 27 25 6F 2D 25 24 3F 3F
6040- 27 00 2D 2D 25 27 25 3F
6048- 3F 2D 2D 24 3C 3F 27 00
6050- 49 25 2C 1F 3E 38 27 24
6058- 6C 09 36 26 00 0E 75 75
6060- 2D 2D 2D 2D 2D 05 65 65
6068- 65 04 00 00 00 00 49 49
6070- 29 4D DA DB 3F 3F 3F 3F
6078- 3F 3F DF 9B 00 35 2E 25
6080- 24 00 20 24 0C 36 36 2E
6088- 20 24 04 00 2D 2D 2D 2D
6090- 2D 24 24 24 24 24 1C 1C
6098- E7 3F 17 F7 1E 36 36 36
60A0- 36 6E 49 24 24 2C 2D FD
60A8- DB 3F 67 49 24 24 24 00
60B0- 24 24 24 24 24 24 2C 35
60B8- 24 3F 34 2D 24 00 24 24
60C0- 24 24 24 24 3C 37 24 34
60C8- 2D 24 00 09 3C 0D 05 38
60D0- 3F 27 0D 0D 1C 3F 0C 04
60D8- 00 29 2D 38 3F 27 0D 0D
60E0- 04 F8 1F 04 00 09 25 3F
60E8- 20 07 20 24 35 36 25 24
60F0- 2D 35 36 1E 37 25 00 09
60F8- 25 3F 28 2D 2C 20 24 3F
6100- 3F 37 2D 2D FE 3F 2E 2E
6108- 04 00 0C 0C 24 24 3C 05
6110- 20 24 3C 38 68 49 49 49
6118- 39 1E 1E 1E 1E 1E 1E 1E
6120- 77 71 0E 0E 0E 0E 0E 25
6128- 00 0C 0C 24 24 3C 05 20
6130- 24 E4 07 68 91 AD 15 36
6138- 36 36 76 0E 2D 20 24 24
6140- 24 24 64 0C 04 00 0C 0C
6148- 2D 2D 2D 2D 0C 0C DF DB
6150- 22 24 24 24 24 3F 3F 1E
6158- 1E 4D 49 08 28 2D 0C 0C
6160- 04 00 49 49 49 29 25 24
6168- 24 24 3F 3F 6F 89 92 3F
6170- 3F 3F 3F 07 38 38 20 24
6178- 24 0C 0C 0C 2D 2D 2D AD
6180- 15 04 00 0C 0C 24 24 24
6188- 24 64 0C 0C 2D 2D AD 15
6190- DF DB DB 92 12 2D 2D 75
6198- 0E 0E 36 36 04 00 49 2D
61A0- 2D 2D 2D 28 F8 DB DB 9B
61A8- 1C 1C 24 24 24 0C 0C 3C
61B0- 6F 29 2D 2D 2D 2D 05 28
61B8- F8 DB DB 93 12 15 36 36
61C0- 36 1E 04 00 15 36 2E 35
61C8- 24 2C 15 36 76 25 24 2C
61D0- 35 36 2E 25 24 2C 35 36
61D8- 2E 05 24 64 35 36 2C 25
61E0- 24 0C 04 00 49 FF
```

武士忠魂

```
1  REM  ************************

3  REM  *         KNIGHT         *

5  REM  *   COPYRIGHT 1985 (C)   *

8  REM  ************************

10  DIM MU(35,2),PE$(16),P(8,8): FOR
     I = 0 TO  PEEK (78) / 5:A =
      RND (1): NEXT : IF  PEEK (2
     45760) <  > 21 THEN  PRINT  CHR$
     (4);"BLOAD KNIGHT SHAPES"
20  HGR : POKE  - 16302,0: SCALE=
     1: ROT= 0: POKE 232,0: POKE
     233,96: HCOLOR= 3: FOR I = 7
     70 TO 790: READ J: POKE I,J:
      NEXT : FOR I = 1 TO 24: READ
     A,X,Y: DRAW A AT X + 1,Y: NEXT
     : FOR I = 24 TO 68 STEP 4: DRAW
     7 AT I,97: NEXT : FOR I = 1 TO
     7: READ G,A,X,Y: HPLOT G + 1
     ,A TO X + 1,Y: NEXT
30  FOR I = 1 TO 6: DRAW I + 15 AT
     I * 15 - 13,50: NEXT : FOR I
      = 1 TO 35: READ A,B: IF A =
     0 THEN  FOR J = 1 TO B * 32:
      NEXT : GOTO 50
40  POKE 768,A: POKE 769,B * 32: CALL
     770
50  FOR J = 1 TO 90: NEXT : NEXT
     : HPLOT 0,0 TO 0,191 TO 93,1
     91 TO 93,0 TO 0,0: FOR I = 1
      TO 185 STEP 23: HPLOT 95,I TO
     279,I: NEXT : FOR I = 95 TO
     279 STEP 23: HPLOT I,2 TO I,
     184: NEXT
60  FOR J = 1 TO 16: READ H,V,G,P
     :N =  INT ( RND (1) * 4 + 1)
     :P(H,V) = J:PE$(J) =  STR$ (
     H) +  STR$ (V) +  STR$ (G) +
      STR$ (P) +  STR$ (N): GOSUB
     430: NEXT :GG = 8:V = 1:H =
     1
70  POKE  - 16368,0:HH = H * 23 +
     74:VV = V * 23 - 20
80  HCOLOR= 0: HPLOT HH,VV TO HH +
     19,VV TO HH + 19,VV + 19 TO
     HH,VV + 19 TO HH,VV: ON  PEEK
     ( - 16384) > 127 GOTO 90: HCOLOR=
     3: HPLOT HH,VV TO HH + 19,VV
      TO HH + 19,VV + 19 TO HH,VV
      + 19 TO HH,VV: ON  PEEK ( -
     16384) > 127 GOTO 90: GOTO 8
     0
90  IF P(H,V) = 0 THEN  HCOLOR= 0

100  IF P(H,V) THEN  HCOLOR= 3
110  HPLOT HH,VV TO HH + 19,VV TO
     HH + 19,VV + 19 TO HH,VV + 1
     9 TO HH,VV: GET A$: IF A$ =
     "I" AND V > 1 THEN V = V - 1
     : GOTO 70
120  IF A$ = "J" AND H > 1 THEN H
      = H - 1: GOTO 70
130  IF A$ = "K" AND H < 8 THEN H
      = H + 1: GOTO 70
140  IF A$ = "M" AND V < 8 THEN V
      = V + 1: GOTO 70
150  IF A$ = "I" THEN V = 8
160  IF A$ = "J" THEN H = 8
170  IF A$ = "K" THEN H = 1
180  IF A$ = "M" THEN V = 1
190  ON A$ =  CHR$ (13) GOTO 200:
      GOTO 70
200  IF P(H,V) = 0 OR P(H,V) > GG
      OR P(H,V) < GG - 7 THEN  GOSUB
     460: GOTO 70
210 H2 = H:V2 = V: POKE  - 16368,
     0:PE = P(H,V):G =  VAL ( MID$
     (PE$(PE),3,1)):P =  VAL ( MID$
     (PE$(PE),4,1)):N =  VAL ( MID$
     (PE$(PE),5,1)): FOR I = 1 TO
     4:MP(I,1) = 0:MP(I,2) = 0: NEXT
     : FOR J = 1 TO N
220  GET A$: IF A$ = "I" AND V >
     1 THEN V1 = V - 1:H1 = H
230  IF A$ = "J" AND H > 1 THEN H
     1 = H - 1:V1 = V
240  IF A$ = "K" AND H < 8 THEN H
     1 = H + 1:V1 = V
250  IF A$ = "M" AND V < 8 THEN V
     1 = V + 1:H1 = H
260  IF A$ =  CHR$ (27) THEN H =
     H * 23 + 74:V = V * 23 - 20:
      HCOLOR= 0: FOR J = V TO V +
     19: HPLOT H,J TO H + 19,J: NEXT
     :V = V2:H = H2: GOSUB 430: GOTO
     70
270  ON H1 = 0 GOTO 220: IF ((P(H
     1,V1) > GG AND GG = 8) OR (P
     (H1,V1) < 9 AND P(H1,V1) > 0
      AND GG = 16)) AND J = N THEN
     J = 0: GOTO 300
280  ON P(H1,V1) <  > 0 GOTO 440:
     MP(J,1) = H1:MP(J,2) = V1: IF
     J > 1 THEN  FOR K = 1 TO J -
     1: ON MP(K,1) = H1 AND MP(K,
     2) = V1 GOTO 440: NEXT
290  GOSUB 450:V = V1:H = H1:V1 =
     0:H1 = 0: GOSUB 430: NEXT : GOTO
     310
300  FOR I = 1 TO 10: POKE 768,I *
     2: POKE 769,I: CALL 770: NEXT
     : GOSUB 450: ON P(H1,V1) = 4
      OR P(H1,V1) = 13 GOTO 340:P
     (H1,V1) = 0:H = H1:V = V1
310 N =  INT ( RND (1) * 4) + 1: GOSUB
     430:H1 = 0:V1 = 0:PE$(PE) =
      STR$ (H) +  STR$ (V) +  STR$
     (G) +  STR$ (P) +  STR$ (N):
     P(H2,V2) = 0:P(H,V) = PE: IF
     G = 1 THEN GG = 16:G = 2: GOTO
     330
```

武士忠魂

```
320  IF G = 2 THEN GG = 8:G = 1
330 V = 1:H = 1: GOTO 70
340 H = H1:V = V1: GOSUB 430: CALL
     - 1998: IF P(H1,V1) = 4 THEN
    P = 2
350  COLOR= 15: HLIN 17,25 AT 23:
     HLIN 17,25 AT 24: HLIN 18,2
    4 AT 25: HLIN 18,24 AT 26: HLIN
    19,23 AT 27: HLIN 19,23 AT 2
    8: HLIN 20,22 AT 29: HLIN 20
    ,22 AT 30: PLOT 21,31: FOR I
     = 10 TO 22: HLIN 16,26 AT I
    : NEXT
360  COLOR= 0: ON P = 1 GOTO 370:
     VLIN 12,22 AT 21: HLIN 18,2
    4 AT 17: GOTO 380
370  FOR I = 11 TO 19: HLIN 21,25
     AT I: NEXT : FOR I = 20 TO
    31: HLIN 10,20 AT I: NEXT : FOR
    I = 20 TO 31: HLIN 10,20 AT
    I: NEXT : COLOR= 15: FOR I =
    21 TO 31 STEP 2: PLOT I - (I
     / 2 - 6),I: PLOT I - (I / 2
     - 6),I + 1: NEXT : PLOT 16,
    20
380  POKE  - 16368,0
390  POKE  - 16298,0: ON  PEEK ( -
    16384) > 127 GOTO 400: POKE
     - 16297,0: GOTO 390
400  TEXT : HOME : VTAB 10: PRINT
    "DO YOU WANT TO PLAY AGAIN ?
     ( Y OR N ) ";
410  GET A$: IF A$ = "Y" THEN  RUN

420  ON A$ <  > "N" GOTO 410: HOME
    : END
430  GOSUB 450: HPLOT HH,VV TO HH
     + 19,VV TO HH + 19,VV + 19 TO
    HH,VV + 19 TO HH,VV: DRAW P +
     11 AT HH + 4,VV + 17: DRAW G
      + 13 AT HH + 10,VV + 17: DRAW
     10 AT HH + 2,VV + 17: DRAW 1
     1 AT HH + 17,VV + 17: DRAW N
      AT HH + 7,VV + 8: RETURN
440  GOSUB 450: GOSUB 460:V = V2:
     H = H2: GOSUB 430: GOTO 210
450 HH = H * 23 + 74:VV = V * 23 -
    20: HCOLOR= 0: FOR I = VV TO
    VV + 19: HPLOT HH,I TO HH +
    19,I: NEXT : HCOLOR= 3: RETURN

460  POKE 768,192: POKE 769,180: CALL
    770: RETURN : DATA 173,48,19
    2,136,208,5,206,1,3,240,9,20
    2,208,245,174,0,3,76,2,3,96
470  DATA 6,1,92,6,69,92,5,1,107,
    5,68,107,5,1,142,5,68,142,5,
    1,177,5,68,177,8,71,175,8,80
    ,163,8,85,153,8,17,153,8,10,
    163,8,3,175: DATA 8,71,140,8
    ,80,129,8,85,118,8,17,118,8,
    10,129,8,3,140,9,30,126,9,51
    ,126,9,30,160,9,51,160
480  DATA 0,92,0,177,22,92,22,177
    ,68,92,68,177,90,92,90,177,2
    3,107,67,107,23,142,67,142,2
    3,177,67,177,115,2,115,1,128
    ,1,145,2,115,1,86,1,96,2,128
    ,4,0,2,145,2,145,1,152,1,172
    ,2,145,1,108,1,115,6,0,2,128
    ,2,86,2,86,2,96,1,86,1,76,2,
    96,1,115,1
490  DATA 96,2,0,1,115,1,86,1,108
    ,1,128,1,145,1,152,2,128,2,1
    45,0,1,1,1,1,2,1,1,1,3,1,1,1
    ,4,1,1,2,5,1,1,1,6,1,1,1,7,1
    ,1,1,8,1,1,1,1,8,2,1,2,8,2,1
    ,3,8,2,1,4,8,2,1,5,8,2,2,6,8
    ,2,1,7,8,2,1,8,8,2,1
```

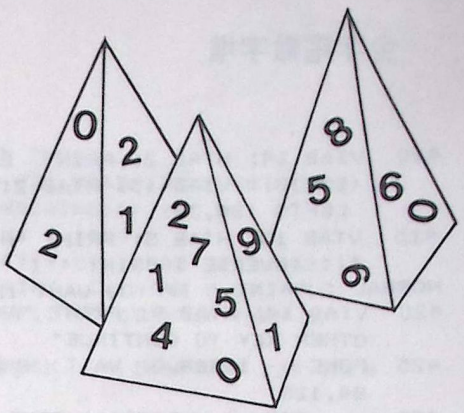
程式開首LIST的特殊效果

GORDEN LUI

許多時程式編寫員都喜歡在前端的幾行用REM 這個指令，而隨後則寫上編寫員的姓名及編寫日期等。相信這種情形讀者們必不會感到陌生。但你們是否想過在列印程式時，開始的幾行只有編寫員的姓名等等資料，而不見了行號及REM 這個指令。其實製造這種效果的方法相當簡單。例如我們想列印時第一行會出現GORDEN LUI 的字樣，我們便應鍵入10REM ：：：：：：：：GORDEN LUI 。這行程式便會在機械碼$801 開始記存。當你鍵入CALL—151 及801L 後，你可看到存於$805的是B2 ，那便是REM 的代號。跟着在$806開始存着八個3A，那是：的ASCII 碼。在行號10中鍵入八個：的目的是在機械碼的相對位置中佔八個空位給我們作更改。我們的工作只需把那八個3A 全改爲08。因08 是代表游標向左移一格，八格之後GORDEN LUI的字樣剛好退至螢幕的最左方；而那些冒號早被08 所代替，已消失於記憶體中。最後需要注意的是，如果冒號後的字數不夠八個，便不能把行號及REM完全遮蓋。■

時間與耐性的賽跑競賽

金字塔數字堆



這是一個跟時間與耐性挑戰的遊戲，誰沉不住便會失敗。玩這個遊戲需要速度、超人的記憶力及相當程度的準確性，一次失手，遊戲便會終結。不過，你不必害怕接受挑戰，遊戲有多種程度開始，由新手至超人程度。

遊戲是怎樣玩法呢？顧名思義，金字塔數字堆是一個 數字 遊戲，但並非要你估數字。遊戲首先會由一個數字開始，它會首先出現，然後你要按入跟著一樣的數字，按對了便算過了一關。不要在此妄下結論，這遊戲斷不會如此簡單。過完一關後，這二排的數字又會閃現，今次有二個數字，然後是第三關的三個字，第四關四個字……數字串的長度會一直增加下去，直至超越你的記憶能力爲止。只要你一次失手，遊戲便完結，需要從頭開始。或者，在限定時間內，如你未能成功按入相同數字，你一樣輸掉！

金字塔數字堆遊戲共有九個難度供你選擇，最後第九個難度只有十秒鐘時間讓你完成每行輸入，假如你能成功地過關，筆者肯定你是天才，因爲連筆者也未能在此關取得滿意的成績。

遊戲的難度只是分開三個部份：玩者需要的時間、每行數字閃現的時間長短、開始分數。難度越高，玩的時間越短，由第一關的九十秒至第九關的十秒。每行數字閃現的時間亦隨難度的增加而減少。開始分數由第一關的130分，以20分的遞增至第九關的最高290分。每完成一行可取得十分。遊戲進行時，可要求數字重現，但每重現一次，要從積分扣廿五分，不要輕率重看數字！

遊戲開始時，你會看到左上角出現的一堆東西，包括說明了現時最高分紀錄，遊戲的難度，及金字塔進行至那一層，最後是現時取得的積分。要數字堆重現一次，可按ESC鍵。

按入數字時，每按一字，鍵盤會有"嗶"一聲，表示經已接受輸入，當按入最後一個數字時，電腦會自動紀錄，故無須按RETURN 或任何其他鍵。祝閣下好運！

本遊戲只有一個程式，檔案名字爲「PYRAMID」。輸入方法很簡單。

```
0  REM  ** PYRAMID **
1  HOME : VTAB 8: HTAB 12: PRINT
     "   COMPUHOUSE   "
2  VTAB 10: HTAB 16: PRINT "PRESE
   NTS"
3  VTAB 12: HTAB 12: INVERSE : PRINT
     " PYRAMID MATCH ": NORMAL
4  FOR I = 1 TO 500: NEXT I
5  VTAB 15: HTAB 12: PRINT "BY
   COMPUHOUSE "
6  VTAB 17: HTAB 14: PRINT "
            "
7  VTAB 19: HTAB 10: PRINT "
                  "
8  VTAB 23: HTAB 22: PRINT "(C)CO
   PYRIGHT 1985"
9  FOR I = 1 TO 4000: NEXT I
10  REM    ** PYRAMID MATCH  **
20  REM
50  HOME : DIM A$(20),AA$(20)
55  FOR I = 1 TO 40:B$ = B$ + " "
    : NEXT
75 DH$ = "-": FOR I = 1 TO 40:DH$
     = DH$ + "-": NEXT I
80 AS$ = "*": FOR I = 1 TO 39:AS$
     = AS$ + "*": NEXT I
85  GOSUB 3600
90 A4$ = AS$
94  GOTO 250
95 PM$ = " < <  P Y R A M I D   M
     A T C H  > > "
100  VTAB 8: HTAB 1: PRINT A4$
110  VTAB 16: HTAB 1: PRINT A4$
120  FOR W = 9 TO 15
125  VTAB W: HTAB 1: PRINT "*": VTAB
     W: HTAB 40: PRINT "*": NEXT
     W
200  VTAB 14: HTAB 8: PRINT "(HIT
      ANY KEY TO CONTINUE)"
210  VTAB 11: HTAB 16: PRINT " PY
     RAMID "
215  FOR I = 1 TO 350: NEXT
220  VTAB 11: HTAB 16: INVERSE : PRINT
     "  MATCH  ": NORMAL
225  FOR I = 1 TO 350: NEXT
230 X =  PEEK ( - 16384): POKE  -
     16368,0: IF X <  = 127 THEN
     210
250  HOME
275  VTAB 1: HTAB 1: PRINT PM$
280  VTAB 10: HTAB 1: PRINT A4$: VTAB
     17: HTAB 1: PRINT A4$
290  FOR FG = 11 TO 16: VTAB FG: HTAB
     1: PRINT "*": VTAB FG: HTAB
     40: PRINT "*": NEXT FG
```

金字塔數字堆

```
 410  VTAB 14: HTAB 2: PRINT  LEFT$
      (B$,38): VTAB 15: HTAB 2: PRINT
       LEFT$ (B$,38)
 415  VTAB 13: HTAB 5: PRINT "HIT
      ";: INVERSE : PRINT "'I'";:
NORMAL : PRINT " IF YOU WANT INSTRUCTIONS;"
 420  VTAB 14: HTAB 9: PRINT "ANY
      OTHER KEY TO CONTINUE"
 425  POKE  - 16368,0: WAIT  - 163
      84,128
 430 X =  PEEK ( - 16384): POKE  -
      16368,0
 440  IF X <  > 201 THEN 1000
 500  REM  ** INSTRUCTIONS
 510  HOME
 515  VTAB 3:C$ = "I'M GLAD YOU AS
      KED . . . ": GOSUB 3570
 517  VTAB 6
 520 C$ = "THE COMPUTER WILL PRINT
       LONGER": GOSUB 3570
 525 C$ = "AND LONGER NUMBERS ON T
      HE SCREEN ..": GOSUB 3570
 530 C$ = "BUT JUST FOR AN INSTANT
      !!": GOSUB 3570
 550 C$ = "YOUR JOB IS TO RE-ENTER
       THEM CORRECTLY": GOSUB 3570

 560 C$ = "AND AS QUICKLY AS YOU P
      OSSIBLY CAN,": GOSUB 3570
 565 C$ = "IN AN EVER-INCREASING P
      YRAMID.": GOSUB 3570
 575  VTAB 22:C$ = "HIT ANY KEY FO
      R THE REST !!": GOSUB 3570: VTAB
      23: HTAB 39
 580 A$ = "": GET A$: IF A$ = "" THEN
      580
 585  HOME
 590  VTAB 6:C$ = "THERE ARE 17 LE
      VELS IN THE PYRAMID.": GOSUB
      3570
 600 C$ = "EACH LEVEL IS HARDER TH
      AN THE ONE BEFORE": GOSUB 35
      70
 602 C$ = "WHEN YOU ENTER NUMBERS
      DON'T WORRY": GOSUB 3570
 604 C$ = "ABOUT HITTING THE RETUR
      N KEY ": GOSUB 3570
 605 C$ = "THE COMPUTER WILL DO TH
      IS FOR YOU.": GOSUB 3570
 615 C$ = "JUST ENTER THE NUMBERS
      THEMSELVES.": GOSUB 3570
 620  VTAB 22:C$ = "HIT ANY KEY FO
      R THE REST !!": GOSUB 3570: VTAB
      23: HTAB 39
 630 A$ = "": GET A$: IF A$ = "" THEN
      630
 650  HOME
 660  VTAB 6
 670 C$ = "YOU CAN ALWAYS ASK THE
      COMPUTER": GOSUB 3570
 680 C$ = "TO SHOW YOU THE NUMBER
      AGAIN, ": GOSUB 3570
 685 C$ = "AS MANY TIMES AS YOU WA
      NT, BY JUST ": GOSUB 3570
 690 C$ = "HITTING THE ESC KEY.  T
      HIS WILL ": GOSUB 3570
 700 C$ = "CANCEL ANY DIGITS ALREA
      DY ENTERED": GOSUB 3570
 705 C$ = "FOR THAT LEVEL, AND YOU
       MAY START": GOSUB 3570
 710 C$ = "THAT LEVEL OVER": GOSUB
      3570
 715 C$ = "BEWARE. . STARTING OVER
       LOSES POINTS!": GOSUB 3570
 720  VTAB 22:C$ = "HIT ANY KEY FO
      R THE REST !!": GOSUB 3570: VTAB
      23: HTAB 39
 730 A$ = "": GET A$: IF A$ = "" THEN
      730
 750  HOME
 775  HOME
 780  VTAB 6:C$ = " THERE ARE 9 DE
      GREES OF DIFFICULTY": GOSUB
      3570
 790 C$ = "THE HIGHER THE DEGREE T
      HE LESS TIME": GOSUB 3570
 800 C$ = "YOU HAVE OVERALL, AND T
      HE LESS TIME": GOSUB 3570
 805 C$ = "YOU WILL SEE EACH NUMBE
      R": GOSUB 3570
 810 C$ = "THE HIGHER THE LEVEL AN
      D DEGREE": GOSUB 3570
 820 C$ = "OF DIFFICULTY, THE MORE
       POINTS YOU GET!": GOSUB 357
      0
 830  PRINT : PRINT
 850 C$ = "HIT ANY KEY TO BEGIN TH
      E CHALLENGE": GOSUB 3570: VTAB
      23: HTAB 20
 880 A$ = "": GET A$: IF A$ = "" THEN
      880
1000  HOME
1004 CT = 0
1005  VTAB 1: HTAB 1: PRINT PM$
1010  VTAB 3: HTAB 3: INVERSE : PRINT
      "PLEASE SELECT A LEVEL OF DI
      FFICULTY": NORMAL
1012 X = 5
1015  FOR I = 1 TO 9: VTAB X: HTAB
      7:X = X + 2
1025  PRINT I;"..";LV$(I): IF I =
      9 THEN  HTAB 10: PRINT "CIVI
      LIZATION"
1035  NEXT I
1050  VTAB 24: HTAB 20
1055 A$ = "": GET A$: IF A$ = "" THEN
      1055
1060 A =  VAL (A$): IF A < 1 OR A
       > 9 THEN 1050
1065 X = 3 + (2 * A): VTAB X: HTAB
      2: FLASH : PRINT " --> ";A;"
      . ";LV$(A): IF A = 9 THEN  VTAB
      X + 1: HTAB 10: PRINT "CIVIL
      IZATION"
```

```
1070  NORMAL : FOR I = 1 TO 2000:
      NEXT I:NS = (10 - A) * 10:D
     F = A
1090  HOME : VTAB 1: HTAB 1: PRINT
     PM$
1100  VTAB 7: HTAB 5: PRINT  LEFT$
     (AS$,30):L$ = LV$(A): IF A =
     9 THEN L$ = "VISITOR"
1105  FOR I = 1 TO 8: HTAB 5: PRINT
     "*"; LEFT$ (B$,28);"*": NEXT
     I: HTAB 5: PRINT  LEFT$ (AS$
     ,30)
1110  VTAB 9: HTAB 18: PRINT "OK,
     "
1112  VTAB 11:L =  LEN (L$):L =  INT
     ((30 - L) / 2): HTAB L + 5: INVERSE
     : PRINT L$: NORMAL
1115  VTAB 13: HTAB 7: PRINT "YOU
      HAVE EXACTLY ";: INVERSE : PRINT
     NS;" SECONDS": NORMAL
1120  VTAB 15: HTAB 9: PRINT "TO
     COMPLETE THE PYRAMID"
1125  VTAB 20: HTAB 5: PRINT "HIT
      ANY KEY TO START THE GAME "
     ;
1130 A$ = "": GET A$: IF A$ = "" THEN
     1130
1150  HOME : VTAB 1: HTAB 2: PRINT
     PM$
1170  VTAB 2: HTAB 1: PRINT  LEFT$
     (AS$,39);
1175  FOR I = 1 TO 21: VTAB 2 + I
     : HTAB 1: PRINT "*";: HTAB 3
     9: PRINT "*";: NEXT I: HTAB
     1: PRINT  LEFT$ (AS$,39);
1180  VTAB 21: HTAB 10: PRINT  LEFT$
     (AS$,21)
1185  HTAB 10: PRINT "*"; LEFT$ (
     B$,19);"*";
1200  GOSUB 5000
1205  PRINT  CHR$ (7): PRINT  CHR$
     (7)
1207  VTAB 24: HTAB 1: PRINT  LEFT$
     (B$,38);: VTAB 22: HTAB 14: PRINT
      LEFT$ (B$,10)
1210  FOR Q = 1 TO 17
1211 T$ = "":TT$ = "":T1$ = "":T2
     $ = "":G$ = ""
1214  VTAB 5: HTAB 8: PRINT Q;
1215  FOR R = 1 TO  LEN (A$(Q))
1220 T$ =  MID$ (A$(Q),R,1) +  CHR$
     (32)
1230 TT$ = TT$ + T$: NEXT R
1240 AA$(Q) = TT$
1250  VTAB Q + 2:HL =  LEN (TT$):
     H =  INT ((40 - HL) / 2) + 1
     : HTAB H: PRINT TT$
1255  FOR I = 1 TO (1500 + ((DF -
     9) * 100)): NEXT
1257 CT = CT + 700
1260  VTAB Q + 2: HTAB H: PRINT  LEFT$
     (B$,HL)
1265  GOSUB 7000
1268  VTAB 22: HTAB 12: PRINT "TI
     ME REMAINING:";
1269 NT = NS * 1000
1270 CT = CT + 58: VTAB 22: HTAB
     27: PRINT  INT (NS - CT / 10
     00);" ";: IF CT > NT THEN 89
     90
1271  IF  PEEK ( - 16384) < 128 THEN
     1270
1272 X% =  PEEK ( - 16384) - 128:
      POKE  - 16368,0
1274  IF X% < 0 THEN 1270
1275  IF X% = 27 THEN  GOSUB 5990
     : GOTO 1211
1276  IF X% < 48 OR X% > 57 THEN
     1270
1277  FOR W = 1 TO 7:WW =  PEEK (
      - 16336): NEXT
1278  PRINT  CHR$ (7)
1279 CT = CT + 180
1280 A$ =  CHR$ (X%):G$ = G$ + A$
     : IF  LEN (G$) = Q THEN 1295

1290  GOTO 1270
1295 T1$ = "":T2$ = ""
1300  FOR R = 1 TO  LEN (A$(Q))
1305 T1$ =  MID$ (G$,R,1) +  CHR$
     (32):T2$ = T2$ + T1$
1310  NEXT R
1315  VTAB Q + 2: HTAB H: PRINT T
     2$
1320  IF G$ = A$(Q) THEN  PRINT  CHR$
     (7): GOSUB 6020
1330  IF G$ <  > A$(Q) THEN  VTAB
     22: HTAB 11: FOR I = 1 TO 5:
      PRINT  CHR$ (7);: NEXT : PRINT
     " WRONG...YOU LOSE!";: GOTO
     9000
1500 G$ = "": NEXT Q
1505 GAME = SC
1510  VTAB 22: HTAB 11: PRINT "
     YOU DID IT!   ": FOR X = 1 TO
     1500: NEXT : GOTO 1000
3570  PRINT :CL =  LEN (C$): IF C
     L >  = 39 THEN  PRINT C$: RETURN

3575 CL = (40 -  LEN (C$)) / 2: PRINT
      LEFT$ (B$,CL) + C$
3580  RETURN
3600  FOR I = 1 TO 9: READ LV$(I)
     : NEXT I: RETURN
3605  DATA  "DROOLING INFANT","LI
     TTLE LEAGUER"
3610  DATA  "JUNIOR HIGH","ALMOST
      DRAFTED"
3615  DATA  "TAKING FRESHMAN MATH
     ", "SUMMA CUM LAUDE"
3620  DATA  "WHIZ KID","COMPUTER
     GENIUS"
3630  DATA   "VISITOR FROM AN ADV
     ANCED"
```

金字塔數字堆

```
5000  REM  ** RANDOM NUMBER GENER
     ATOR AND PARSER
5005 NT$ = ""
5010  FOR J = 1 TO 153
5015 N$ =  RIGHT$ ( STR$ ( INT ( RND
     (1) * 10)),1)
5020  IF J = 1 AND N$ = "0" THEN
     5015
5030 NT$ = NT$ + N$: VTAB 22: HTAB
     19:J$ = "    " +  STR$ (154 -
     J): PRINT  RIGHT$ (J$,3): NEXT

5050 PK = 1
5060  FOR K = 1 TO 17: VTAB 24: HTAB
     8: FLASH : PRINT "WATCH SCRE
     EN CLOSELY PLEASE ";: NORMAL

5065  IF  MID$ (NT$,PK,1) = "0" THEN
     NT$ =  LEFT$ (NT$,PK - 1) +
     "1" +  MID$ (NT$,PK + 1)
5070 A$(K) =  MID$ (NT$,PK,K)
5080 PK = PK + K
5090  NEXT K
5100  FOR I = 1 TO 3: VTAB I + 2:
      HTAB 14: PRINT "!";: HTAB 2
     6: PRINT "!";: NEXT : VTAB 6
     : HTAB 13: PRINT "/";: HTAB
     27: PRINT  CHR$ (92);
5105  VTAB 7: HTAB 2: PRINT  LEFT$
     (DH$,11);: HTAB 28: PRINT  LEFT$
     (DH$,11);
5110  VTAB 3: HTAB 2: PRINT "HI G
     AME:";: PRINT GAME;
5115  IF GAME = 0 THEN  HTAB 10: VTAB
     3: PRINT "---";
5120  VTAB 4: HTAB 2: PRINT "DIFF
     ICULTY:";DF;
5130  VTAB 5: HTAB 2: PRINT "LEVE
     L:";
5140  VTAB 6: HTAB 2: PRINT "SCOR
     E:";
5150  VTAB 3: HTAB 27: PRINT "ENT
     ER NUMBER";
5160  VTAB 4: HTAB 27: PRINT "AFT
     ER FLASH.";
5170  VTAB 5: HTAB 27: PRINT "<ES
     C> STARTS";
5180  VTAB 6: HTAB 29: PRINT "LEV
     EL OVER";
5200  RETURN
5990 SC = SC - 25
6000  VTAB 6: HTAB 8: PRINT SC;"
     ";: RETURN
6020 SC = SC + 100 + Q * 10 + 2 *
     (100 - NS): VTAB 6: HTAB 8: PRINT
     SC;" ": RETURN
7000  IF CT / 1000 > NS THEN  VTAB
     22: HTAB 12: PRINT "   OUT O
     F TIME!  ";: PRINT  CHR$ (7)
     ;: GOTO 9000
7010  VTAB 22: HTAB 12: PRINT "TI
     ME REMAINING:" INT (NS - (CT
      / 1000));" ";: RETURN
8000  HOME : END
8990  VTAB 22: HTAB 12: PRINT "
      OUT OF TIME! ";: PRINT  CHR$
     (7);
9000  FOR X = 1 TO 2200: NEXT X
9020  VTAB 22: HTAB 11: PRINT "
     R=REPLAY Q=QUIT ";: GET A$
9040  IF A$ = "R" THEN 9075
9050  IF A$ = "Q" THEN 8000
9060  GOTO 9020
9075  IF SC > GAME THEN GAME = SC

9080 SC = 0: GOTO 1000
```

有保護遊戲之拆圖方法

呂樹偉

你試過BOOT了一隻遊戲碟，又見到那遊戲的繪圖非常美麗，想把那圖拆出來，BSAVE做FILE，以後慢慢欣賞，我也嘗試過。我想一隻遊戲碟，在BOOT機之後，已經控制了3F2和3F3，只要按下CTRL RESET，立即BOOT機或LOCK機，關鍵在於CTRL—RESET那處，查了很多書，終於找到了，除了RAM上的3F2，3F3這兩個位置和RESET有關係之外，在ROM上也有兩個位置FFFC和FFFD。而我又發現那兩個位置存的值是$62和$FA，我試試FA62G竟和RESET一樣，問題在於ROM上的資料怎樣更改呢？用16K RAM咭！大家可先CALL—151，再照下列方法做：

C081 C081（使RAM咭變成可讀／寫）
D000<D000.FFFFM（把ROM搬上去）
C083 C083（使RAM咭變爲可寫入）
FFFC：59 FF（存入MONITOR的進入點）
C080（使RAM咭變成只可讀狀態）
0G
PR#6
：：：：（當那圖出現時）
CTRL—RESET
BSAVE PICTURE，A$XXXX，L$XXXX

當然沒有一種方法是萬能的，有些GAME會用RAM咭來存放資料，這方法便行不通了！例如BRUCE LEE便是；但不妨試試！

最新遊戲軟件評價　李德華

WARRIORS

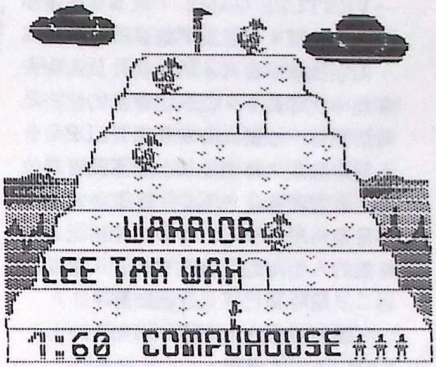
WARRIORS 是一個富有想像力及非常精彩的ACTION GAME ，玩者可選擇LEVEL 來開始或更高的LEVEL開始，而且遊戲更可多至八人進行，在每一個LEVEL 開始前都有詳盡的文字及圖解說明，所以筆者現簡略地介紹每個LEVEL 的玩法。

LEVEL 1：在這LEVEL 中，你會置身在山脚下，在山上有恐龍把守着，你要盡辦法把圖形結合，而你可以用石頭射他們，當拿了五個圖形後，便會有一道七彩繽紛的門在山頂上，只要通過便升LEVEL了。

LEVEL 2：在這LEVEL 中，你會置身在馬場內，而你的目的是要把馬場上的金幣接住，但途中不斷有馬追逐你及馬場上有水漬及石頭襲擊你，那些馬可以用你的智慧把它們摔倒，只要拿到五個金幣便升LEVEL了。

LEVEL 3：在這LEVEL 中，你會置身在一個堡壘外，而且又有士兵來對付你，你要不斷用箭對付他們，而且更要搶那有錢的人的錢及拿老人家的拐杖，直至拿到三枝爲止，然後便升LEVEL。

LEVEL 4：你會置身在內戰中，而你乘搭的是氫氣球，在地上有無數的大砲在攻擊你，你要不斷找尋能源包，當找到了三個能源包，便能升LEVEL了。



LEVEL 5：你會置身在市區中，而你會被一頭狗及一架汽車追逐，汽車可以用小型炸彈將之毀滅，而你要在郵政局外按BUTTON 0，那郵政局便會顯示不同的數字，而畫面亦會有一點東西在閃動，把郵包送到那地方，畫面的左下角便會出現屋的圖形，餘此類推，直至把整間屋顯現出來，然後進入屋裏便行。

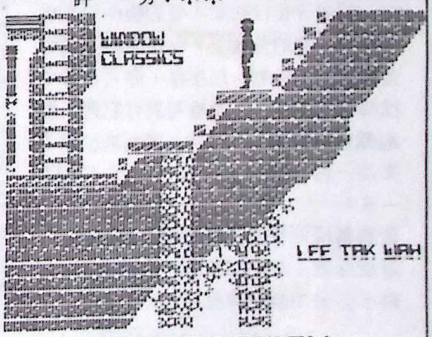
LEVEL 5後，便是BONUS LE—VEL，而你會在懸崖上，只要跨過那山頭，便會有獎分了。經過筆者簡略的解釋，讀者們也應知道是怎樣玩的了。

遊戲名稱：WARRIORS
遊戲類型：ACTION GAME
評　　分：三粒半星

WINDOW CLASS

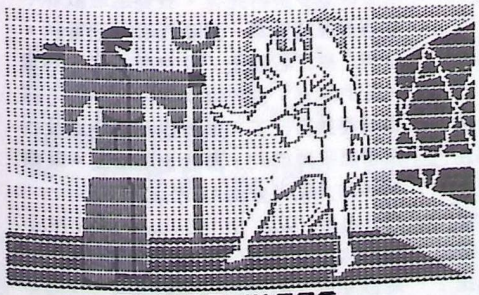
WINDOW CLASSES 是一個AC—TION ADVENTURE GAME，玩者可以在每間屋及間每個人找尋線索，在屋裏的東西，一定要問過主人才能拿，而遊戲中有不同的命令來幫助玩者去調解這遊戲，例如SPEAK 一可以和人說話；SLEEP—睡覺；USE 一用你拿到的東西；EXAMINE 一觀察物件或東西；MENU—LOAD GAME或SAVE GAME，筆者花了不少時間去玩這遊戲，也不知道怎樣才能破解，而且玩的時候是漫無目的，眞是氣死人。

遊戲名稱：WINDOW CLASS
遊戲類型：ACTION ADVENTURE
評　　分：＊＊



GAME REVIEW

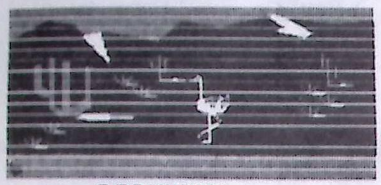
SHADOWKEEP



SHADOWKEEP 是近期最受歡迎的ADVENTURE GAME 之一。整個遊戲共分四面磁碟，這遊戲開始時，會發問很多不同類型的問題，其中一個問題是能夠替用者進行BACK UP，而其中一個問題是玩者必須做的，它就是替玩者FORMAT一張磁碟作SAVE GA—ME 用。完成以上步驟後，遊戲更詢問一連串問題，但這些問題，相信聰明的讀者一看也會明白，筆者在這裏不願多講。

再而便是到商店買武器及必須品，由於你的黃金數目有限，所以並不是每樣也能買到，選擇了武器及必須品後，便正式進行遊戲了，這時電腦會打出轉去第二面碟的訊息，第二面碟主要是一個龐大的迷宮，迷宮內有超過一百個不同的殺手（包括動物），如果不幸被他們包圍，這時便要和他們展開決戰或者撤退，這些壞傢伙並不是容易對付，如果你們有這個遊戲的MENU，相信會更加易掌握遊戲的玩法。

遊戲名稱：SHADOWKEEP
遊戲類型：ADVENTURE
評　　分：＊＊＊



魯賓遜飄流記

魯賓遜飄流記是根據魯賓遜飄流記的故事改編而成的ADVENTURE GA—ME，遊戲設計得非常好，因爲輸入答案的方法再不是全部打入的落後方法，而是打入一個或以上的字，然後按—SPACE 便會把整個字顯示出來，而按／鍵可把所有命令顯示出來。

遊戲的開始是要拿取不同的物件，想辦法移開木桶，跟着便會找到船，把牛、木桶、家人……等的東西運上船上，當你在海上航行的時候，便把船底弄穿，跟着你和你的家人等的物件便會飄流到紐西蘭的荒島上，你要在島上找尋食物，並和動物一起生存，你可在書中找尋動物的資料，用槍可對付它們，用地圖來觀察島上的環境，總括來說也不失是一個好玩的ADVENTURE GAM—E。

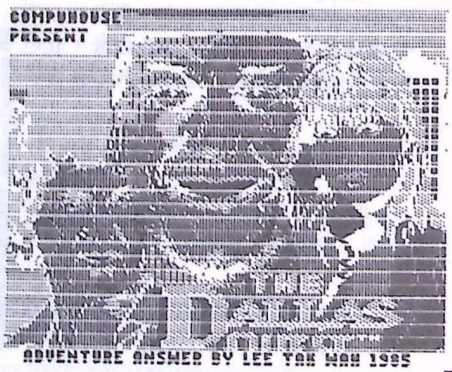
遊戲名稱：魯賓遜飄流記
遊戲類型：ADVENTURE GAME
評　　分：兩粒半星

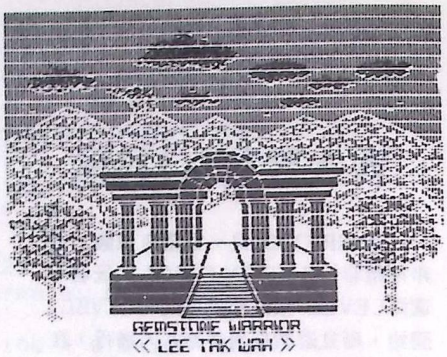
豪門恩怨

豪門恩怨這套長篇電視劇在美國曾哄動一時，而在港也曾播映過。最近坊間就出了一個改編自這套電視劇的AD—VENTURE GAME ，故事是描述你是一個偵探，現在要爲這家破一件奇案，而在故事中會有不同的物件及人物去帮助你找尋線索，但玩這遊戲的秘訣是看清楚每一幅畫及時常利用HELP命令，這遊戲最大特色是加上動畫及音樂效果，令這遊戲生色不少，在遊戲當中，時常要利用OFFER 命令，相信玩過這遊戲的人也知道是怎樣用的了，筆者經過二星期時間已將這遊戲破解。

讀者若想索取答案，可向APPLE FAN CLUB查詢。

遊戲名稱：豪門恩怨
遊戲類型：ADVENTURE
評　　分：＊＊＊





GEMSTONE WARRIORS

GEMSTONE WARRIORS 是一個非常具刺激性的ACTION GAME ，在遊戲中，你是一個寶石戰士，你要在破舊的廟宇當中找尋寶石，但在廟宇分散了不同的怪物來對付你，這些怪物是會把你的能源吸乾，幸而你有飛箭來保護自己，而在廟宇中更有不同的物件來補充，這些物件對於你有莫大帮助，例如有隱形功能、飛箭補充、強力毀滅炸彈、補充能源、保護罩……等。玩者要不斷過不同的門及閘，但每一道門及閘也隱藏了殺機，你有沒有機會拿到寶石，就要看玩者的運氣如何了。

遊戲名稱：GEMSTONE WARRIOR
遊戲類型：ACTION GAME
評　　分：兩粒半星

TRACER SANCTION

近期坊間出了不少ADVENTURE GAME，而推出的ADVENTURE GAME 大多以動畫形式來增加遊戲的效果，是冒險遊戲的大突破。

這個TRACER SANCTION 也是以這種形式來吸引玩者。TRACER SA—NCTION 是一個太空尋寶的冒險遊戲，你要在宇宙中的九個行星中尋找寶藏及破解的線索，只要買足夠的油及乘坐太空船，便能夠縱橫宇宙，可是你往往要找錢買油，但每個星球也有不同的難關教你去面對，以後才能找到錢或物件，而全宇宙最大的寶藏是在地球，但筆者經過多番研究後，發覺這遊戲並不是尋寶那麼簡單，還要找尋一把寶劍，並將之插在石頭上，跟着要過電腦隨意分配的迷宮，由於每次迷宮都不同，這裏就要靠自己了，跟着返回出發地及找那帶黑眼鏡的人，遊戲也就結束。遊戲的答案大家可查詢APPLE FAN CLUB。

遊戲名稱：TRACER SANCTION
遊戲類型：ADVENTURE
評　　分：＊＊＊

F-15 的秘密

在近期有一個新ACTION，比SKY FOX 更好玩，而且武器更多，刺激性更高，相信玩F— 15這個遊戲的讀者，都會買使用手冊，因爲如果沒有使用手冊，根本不能玩這個遊戲，但當買了使用手冊之後，玩起來，又不知怎樣用武器，看熟了使用手冊，才知要看使用手冊中的TOP SECRET內的CODE ，才能用武器，但一看使用手冊中的TOP SECRET ，嘩！不知是甚麼，看不清楚，所以在此和讀者分享一下。

一開始的時候，會問你有幾多個人玩和去攻擊甚麼，按了BUTTON 0之後，會叫你入密碼，電腦會出不同的數字叫你入不同的英文字母。如果它出0，你就一定要入G，1入E，2是A，3是M，4是C，5是D，6是F，7是E，8是M，9是P，10是J，11是C，12是E，13是C，14是B，15是M。

當入完了密碼之後，如果有使用手冊的話，不妨看看以下的武器用法：

按0－9是選擇機器的能力（POWER）

ESC 是跳出遊戲到最開始的時候。

R 是選擇雷達，開始是中程雷達，按一次是遠程雷達，按多一次是短程雷達。

E 是用來避開地對空導彈。

A 是輔助引擎。

S 是裝上短程導彈，按BUTTON 0即發射。

X 是SPEED BREAK，是將速度減至最慢。

D 是用來放下油箱，當能源低過13,300的時候，即可以放下，減輕機身，加快速度。

F 是用來避開空對空導彈。

G 是GUN ，當你裝上了其他武器，而而又想變回槍的時候，即可按G。

B 是裝上炸彈，用來打地上的地對空導彈發射場，機場及基地。

M是裝上中程導彈。

SPACE 是用來看機前及機後。

P 是停止遊戲。

K L 是控制地圖上的導向游標，當你按這些制把游標移到你要去的地方，你會看到銀幕上有N，A，V，這三個字母閃動，只要將對準器對準閃動的位置，然後按一BUTTON 0，即會飛到你要去的地方。

希望這小介紹能幫助讀者快點了解F—15 這個玩意。■

捉蟲訓練—

這是一個九乘九的乘數表的程表，但當它運行時，只見螢幕上有一大堆數字。究竟如何能把它排列成一個表呢！程式：

```
10  HOME
20  PRINT "  +   1   2   3   4   5   6
     7   8   9"
30  PRINT "======================
     ======="
40  FOR I = 1 TO 9
50  PRINT I;"+";
60  FOR J = 1 TO 9
70 M = I * J
80  IF M < 10 THEN  PRINT "  ";M;

90  IF M >  = 10 THEN  PRINT " ";
     M;
100  NEXT J
110  NEXT I
```

我們歡迎讀者提供〔捉蟲訓練〕問題。程式長度沒有限制，以越簡短越好。程式必須要以APPLESOFT 寫成，並符合下列各項要求：

1. 程式必須是APPLESOFT ，任何POKE或CALL機械語言的敍述必須正確。
2. 程式必須能出現某些結果，而不可出現SYNTAX ERROR。
3. 所有問題必須附加簡短說明及答案。

假如閣下的作品獲得刊登，我們將會送出十二期免費雜誌給你或將你的訂閱延長一年。

所有投稿請寄九龍中央郵政信箱C. P. O BOX 71193 。信封面請寫上〔投稿捉蟲〕。■

答案：

只要在行句100後處加入「:PRI—NT」便會列出一個表。

WILDCARD能夠解拆保護的原理

蔡紀聰

WILD CARD這個名字在衆多的AP Ⅱ用家心目中只是一塊用來拷貝程式那麼簡單的週邊咭。例如一個機械語言程式執行時你可以利用WILD CARD隨時檢知A、X、Y、SP、PC 各個暫存器的內含值，這樣不單對於捉蟲及了解程式的運作有幫助，對於碟及帶的解拆更加有用，WILD CARD還可以將程式暫停下來讓使用者可以觀看和修改記憶內的資料或程式，然後將被中斷的程式繼續執行，例如玩GAME時可以利用WILD CARD去增加玩者的機會。

```
****************
*  WILD  CARD *
* A$1000,L$2D *
****************
1000- 20 58 FC BA A9 10 9D 00
1008- 01 CA A9 1B 9D 00 01 CA
1010- A9 4C 9D 00 01 A9 01 85
1018- 4F 86 4E 2C 10 C0 AC 00
1020- C0 8C 00 04 2C 30 C0 88
1028- D0 FD 6C 4E 00 F0
```

內部運作過程

事實上WILD CARD拷貝程式的過程是相當簡單，首先WILD CARD 使到6502 產生硬體的插斷（INTERRUPT），當6502 發生硬體插斷時，6502 便會立即自動地將當時程式計數器（PROGRAM COUNTER）及狀態暫存器（STATUS REGISTER）的內含值存於當時堆疊指標（STACK POINTER）所指的堆疊位置內，然後6502 便會去執行WILD CARD上的程式，而這程式的主要作用是將累加器（ACCUMULATOR）、X 暫存器（X REGISTER）、Y 暫存器（Y REGISTER）及堆疊指標（STACK POINTER）在中斷時的內含值存於RAM中，然後將記憶體內（$0000至$3FFF）的資料搬移上RAM CARD，跟著就執行WILD CARD內的主菜單。以上就是WILD CARD運作過程中最重要的部份。

至於RESTART方面，當執行WILD CARD 主菜單上的RESTART時，RESTART 程式將內裏的資料放回（$0000至$3FFF）位置，然後將堆疊指標、累加器及X 和Y 暫存器的內含值還原，再用6502內的RTI 指令將程式計數器及狀態暫存器從堆疊指標所指的堆疊位置內將其原本的內含值取回，當程式計數器的內含值還原後，6502 便去到程式計數器的內含值所指的位置去繼續執行程式。

檢查各暫存器的內含值

首先按下WILD CARD上的鍵後按下RETURN鍵，然後按M鍵進入MONITOR 內，這時累加器的內含值便存於$FFC內，而X 暫存器的內含值存於$FFD 內，Y 暫存器的內含值存於$FFE內，堆疊指標的內含值存於$FFF內，而狀態暫存器及程式計算器的內含值存於堆疊指標內含值（$FFF）所指的堆疊位置上，例如堆疊指標內含值是$FC，因此狀態暫存器的內含值便存於位置$1FC＋1內，而程式計算器的內含值便存於位置$1FC＋2（低位元）及$1FC＋3（高位元）內。各位可以在記憶體中直接修改各暫存器的內含值，在這裏要特別一提大家的是，如果各位要修改記憶（$0000至$3FFF）內的資料時，由於WILD CARD已將（$0000至$3FFF）內的資料移上RAM CARD，因此要修改在這範圍內的資料時，必須去修改RAM CARD上的資料，而不是直接修改（$0 至$3FFF）內的資料，例如要修改位置$0內的資料爲＃$FF方法就是按C089 C089 D000：FF，要修改位置$1000內的資料爲＃$00其方法就是按C081 C081 D000：01 ，要修改位置2000內的資料爲＃$7F 其方法就是按C081 C081 E000：7F 。

缺點

WILD CARD 的缺點數起來已不少。不能拷貝一個足64K 的程式，和拷貝速度慢等都是WILD CARD的缺點，但它的優點始終都比缺點多，因此它才可以在衆多AP Ⅱ用者中流行起來。

但WILD CARD還有一個很少人知道的缺點，這個缺點的形成是由於WILD CARD利用6502 的硬體插斷所至，當6502 的硬體插斷發生時，6502 立即將狀態暫存器及程式計算器的內含值存於堆疊內，因此堆疊內便有3個位元組內的資料被狀態暫存器及程式計數器的內含值所取代。聰明的人便利用WILD CARD這個缺點來保護自己的程式不被WILD CARD所拷貝，如LOCKSMITH 5.0 爲內的FAST COPY （FAST COPY 並沒有應用到RAM CARD）爲例，當FAST COPY BOOT 起後，各位讀者試用你的WILD CARD將它拷貝看看是否成功。下面有一個不足一頁長的機械碼程式，這個程式執行後用者在鍵盤上按下不同的鍵便會有不同的聲音發出，各位讀者試試用WILD CARD將它拷貝看看是否拷貝成功。（這個發聲程式並沒有用到RAM CARD）大家亦可從今期的程式磁碟中找到這個程式。■

WILD CARD與RAM咭的配合應用

孫悟空任你玩

CONAN是一隻十分刺激好玩的遊戲，但是和TAPPER 一樣是相當難玩，因此小弟在這裏教各讀者如何改人數、選LEVEL 、加飛刀。出術的工具也和以前一樣，包括RAM CARD 及WILD CARD各一塊，當工具準備好後便將CONAN BOOT起直至見到LEVEL 1，這時你便按下連接在WILD CARD上的鍵，然後小心地按下RETURN 鍵，畫面上便會出現了WILD CARD的功能表，這時按下M鍵，便會在畫面上出現一個＊（即是乘號），這時你便依照下面的步驟按鍵：

①C089 C089 〔RETURN〕

②D34B：

36 05 09 09〔RET〕

人數54 LEVEL 5 飛刀99張

③C050 C054 C057 C052 B77G 〔RETURN〕

你現在已有54個人，99張飛刀，當你過了LEVEL 1後下一個LEVEL就是LEVEL 6 。上面的修改人數最多是54人，飛刀最多是99張，而LEVEL 可以是1至7。■

抽不死的啤酒佬

啤酒佬（TAPPER）這隻GAME是講述一班酒鬼去酒吧裏飲酒的情形，這班酒鬼十分之兇惡，假如你不能迅速將啤酒送到這班酒鬼手上的話，可能連頭都被這班酒鬼打破。

TAPPER 是一隻很好玩的GAME，但它也是一隻很難玩的GAME。本人未曾見過有一個人能眞眞正正地玩到上太空，對於這樣難玩的GAME，要玩完所有LEVEL 就要花少少心思，才可以有更多的機會去玩這個GAME。出術的工具包括RAM CARD及WILD CARD各一塊，方法首先將TAPPER BOOT起，當畫面出現酒鬼後便按下WILD CARD 上的鍵，跟着再照下面去按鍵就行：

①按〔RETURN〕

②按M

③按C089 C089〔RETURN〕

④按D06E：7E 〔RETURN〕

⑤按C050 C057 C052 B77G 〔RETURN〕

你現在已有128次機會，如果128次的機會還不能令你上到太空你可以在機會接近用完時，再用上面的方法修改，之後你便會重新得到128次的機會，努力吧！有朝一日你定能上到太空。■

・蔡紀聰・

打不死的RUN FOR IT

以下的方法可將RUN FOR IT 遊戲「打不死」：

①機械人碰到怪獸和能源都沒有反應。

②能源不會逐漸減少，除非碰到怪物。機械人碰到怪獸和能源都沒有反應

如有WILD CARD，可在遊戲開始時按下WILD CARD的按鈕，然後按〔RETURN〕，跟住按〔M〕進入監督程式。然後依照下列方法做：

・＊C089 〔RETURN〕

・＊C089 〔RETURN〕

・＊D8AD：EA EA EA 〔RETURN〕

・＊C050 C057 C052 N B77G 〔RETURN〕

如沒有WILD CARD不要緊，只要利用NIBBLES AWAY II 便可做到以上的效果。

首先BOOT起NIBBLES AWAY II，然後按〔S〕進入SECTOR EDIT－OR，跟住按〔O〕，然後按〔C〕，再按〔RETURN〕。然後依照下列方法做：

T01 〔RETURN〕

S0E 〔RETURN〕

R

把地址＄AD的值改爲＄EA；地址＄AE改爲＄EA；地址＄AF改爲＄EA

W

能源不會逐漸減少，除非碰到怪物

如有WILD CARD，用以上的方法進入監督程式。然後依照下列方法做：

＊C089 〔RETURN〕

＊C089 〔RETURN〕

＊D8B0：EA EA EA 〔RETURN〕

＊C050 C057 C052 N B77G 〔RETURN〕

如沒有WILD CARD，同樣依照上述方法。然後依照下列方法做：

T01 〔RETURN〕

S0E 〔RETURN〕

R

把地址＄B0的值改爲＄EA；把地址＄B1改爲＄EA；把地址＄B2改爲＄EA

W ■

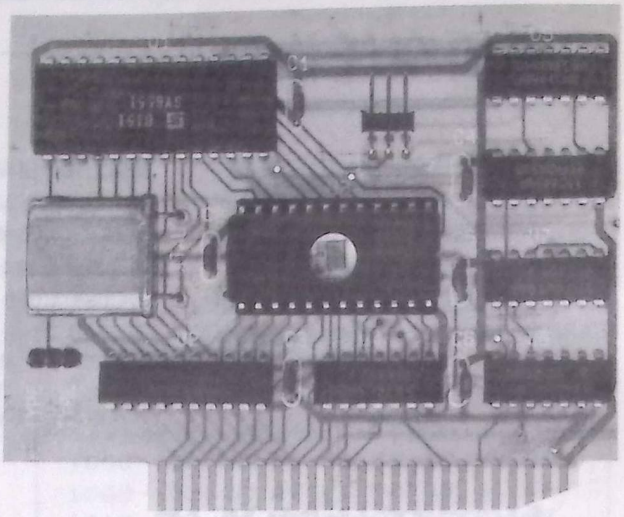
・梁偉强・

專業化電腦硬件維修/改良服務

APPLE II或相容機
—主機（HK$80）
—磁碟機（HK$30）
—一般介面咭（HK$25）
—IBM或相容機（HK$250）

＊上述費用不包括零件費
＊詳細收費服務表備索

本會擁有一群豐富專業知識的電腦工程人員，抱著非牟利的宗旨爲廣大的電腦用家提供廉價而專業化的電腦維修保養／改良及其他服務。



本會現招收基本會員，歡迎參加，章程備索。

電子科技學會 ELECTRONIC TECHNOLOGY ASSOCIATION

收件站：旺角上海街438號同珍商業中心十一字樓四室（電梯十一字）

專改卡仔

徵聘特約撰稿員

任何人士有興趣寫作有關CP/M, Z-80, IBM, dBASE II, WORDSTAR, VISICALC或高等語言文章，或有編寫有關程式經驗，歡迎你加入成爲我們的特約撰稿員，待遇從優。

請附同個人學歷／簡歷，連同個人在電腦興趣方面的專長寄來出版社，地址是：九龍中央郵政信箱71193號。

徵求程式磁碟 BOOTING 程式

電腦時代每期出版的程式磁碟徵求動畫製作的BOOTING程式。

在過去多期的程式磁碟內，大家只要BOOT 起磁碟即會出現一些美麗的圖案，告訴用家是那一期的程式磁碟。

現在，我們徵求一些類似的BOO—TING 程式，放在每一期出版的磁碟上。

BOOTING程式必須要以動畫形式寫成，大家可利用TAKE 1，或GRAP—HIC SOLUTION等動畫製作程式來設計。有關的例子大家可參考第十七期的程式磁碟。

所有投稿設計，必須連同磁碟及簡要說明寄來，我們的通訊地址是：九龍中央郵政信箱71193 號。並請附同一個回郵信封，以便將磁碟退回。

入選的作品將可獲得自由選擇任何一期的程式磁碟一張。此外，我們將會在連續十二期內的入選作品之中，選出最優異的一個，送出高級磁碟一盒（十張）。

所有作品，內容形式不限，但必需要有「COMPUTING AGE PROGR—AM DISK」字樣。■

有興趣而尚未加入APPLE FAN CLUB的朋友，歡迎來信索取有關通訊資料，請連同回郵信封（貼上五角郵票）另付上一元（可用郵票代替），寄來讀者服務部，信封面請註明「索取AFC資料」。■

兩個ADVENTURE GAME答案推出

ADVENTURE 遊戲解破答案
磁碟版本推出
每張訂價20元（連磁碟）

1) INSTITUE & LUCIFER'S RELAM
2) DALLAS QUEST & TRACER SANCTION

TAKE1 電影大獎揭曉

第十五期舉辦的TAKE 1電影大獎比賽，我們一共收到三份參賽作品，經審核後，我們選出金墨田先生的"約會"爲優勝作品。

我們將優勝作品製成磁碟版本公開出售，每張連磁碟十五元。大家可到讀者服務部選擇。■

MOCKING BOARD在本港興起的日子很短，但十分受到歡迎，主要是它爲APPLE 電腦的發聲部份開創了一個全新的領域——從單調的單音道聲頻，轉化出多姿多彩的立體聲音响效果。目前來說，還未有一張硬體週邊，可以比MOCKING BOARD更能發揮到身歷聲的完美音响效果。

MOCKING BOARD除了一塊硬件外，附隨的尚有一本使用手冊及一張—SOUND UTILITIES 的磁碟。

可惜出品MOCKING BOARD的公司，並未能將有關MOCKING BOARD的使用手冊編製得如其硬件一樣的出色，手冊上提供的資料，無論是技術或應用都未能令人滿意。要充份發揮這張硬體，需要花很多精神和時間鑽研，最致命的一點是市面上能提供的有關參考書籍，極之貧乏。

電腦遊戲專欄作者麥法基先生過去所編寫的遊戲軟件皆受到蘋果電腦迷廣大的歡迎，他的作品水準大家都是有目共睹，麥先生出色之處在於能在每個編寫的遊戲中加插許多不同的技巧。而且在多個ACTION GAME ，他將MOCK—ING BOARD 所能做到的音响效果發揮得淋漓盡緻。他對MOCKING BO—ARD 的使用有深切的認識。

鑑於許多電腦用家對MOCKING BOARD的歡愛，而市面又缺乏有關的參考書籍，我們特意邀請麥先生編著一本〔MOCKING BOARD應用技術〕的參考書。書本內容包括五大方面：

(1) MOCKING BOARD 和聲音關係—詳細解釋MOCKING BOARD硬體和軟件結構。
(2) MOCKING BOARD 的基本運用—提供聲音的各種UTILITIES 。
(3) 配合MUSIC CONSTRUCTION SET的應用。
(4) 各種程式實例—把運用方法加入現有程式內及如何以簡單BASIC 編寫程式。
(5) 以繪圖和動畫方法介紹各種機械語言指令的學習。

單看上述所蓋括的內容，我們深信足以令你發揮這張咭的優點。

MOCKING BOARD應用技術一書將於六月底出版，32開印行，每本訂價30元。現接受預訂。凡於五月三十日前寄來預訂表格，可獲特價優待，只需廿五元。請利用下列之預購表格：

本人現欲訂購〔MOCKING BOARD 應用技術〕____ 本。現付上劃線支票No._____(銀行)_____ 一張／滙票（如用郵政滙票請勿劃線）

本人採用下列方式領取（掛號請加四元）
郵寄 □ 親身到讀者服務部領取 □
如果是到取，請寫上身份證號碼。

姓名：_____
地址：_____
_____
電話：_____

以下爲本人之回郵地址：
（請用正楷書寫清楚）

姓名：_____
地址：_____
_____
_____

各期程式磁碟磁帶總目錄

1
1 突破傳統的鍵盤
2 捲球遊戲
3 勇闖電腦室
4 迷宮救美
5 將 APPLESOFT 變爲 FORTRAN
6 上、下、左、右移動文字畫面
7 測試工具— BOOLEAN 程式
8 騰空更多磁碟空間 —搬移HELLO
9 重現刪除之檔案
10 電腦時代月曆
11 COMPUTEXT 文字處理
12 時日相距計算程式
13 增強英文字滙能力
14 幼童數學四則運算練習
15 過四關

2
1 貨櫃處理
2 沙蟹幻方局
3 巧破密令
4 美術字繪劃器
5 小階字母處理
6 程式轉載器
7 彩色電視／監視器 色彩測試
8 保留遊戲積分記錄
9 數學套件程式
10 磁碟修護工具
11 加設電腦時代月曆 圖像一個

3
1 DOS 的自動同步追踪器
2 動量不滅定律模擬法
3 自製磁碟索引表封套
4 蘋果編曲器
5 幼童認數字初階
6 賽跑遊戲
7 步步追踪
8 DOS KILLER
9 圍棋遊戲
10 巴哈的古典樂章名作
11 倒轉螢幕畫面
12 雜誌目錄分類歸檔系統
13 神經病醫生—神經鬆弛體操

4
(1) DOS 操作指令簡化器
(2) TEXT FILE 有些甚麼內容？
(3) 立體迷宮
(4) 英文串字練習
(5) 如何以機械語言來修補 DOS
(6) 電腦動畫製作原理。
(7) 豐富設計的學前兒童教育程式
(8) 應用數學套件程式(二)
(9) DOS WRITER
(10) 穿梭機哥倫比亞號機
(11) 找回失落了的 STEP 及 TRACE
(12) TRACER控制常式
(13) 將 RAM 卡作爲磁碟機使用
(14) 廚房計時器
(15) 太陽系模擬系統

5
1 私人約會備忘日曆
2 三度空間顯像系統
3 將 CP／M 檔案轉爲 DOS
4 蘋果萬花筒
5 高解像度文字產生器
6 修訂 DOS 操作指令簡化器
7 濃縮圖片儲存空間
8 幼童數學練習—數字咭
9 加數機大登場
10 備忘 MEMO 紙
11 光速計數器
12 剩餘磁碟空 SECTOR 數目
13 16進制輸入數字鍵盤
14 問題棋盤
15 神槍智破夾萬
16 隔山打牛
17 大空船降落任務

6
(1) 私人會計系統。
(2) APPLESOFT 擴展版本。
(3) GO] [HELLO 的程式
(4) APPLESOFT 偵錯系統。
(5) DOS 指令再修訂版本。
(6) 布倫時鐘。
(7) AMPERSAND 數基轉換器
(8) 完整索引錄。
(9) 砌積木。
(10) 神奇數字方塊。
(11) 亡命飛車。
(12) 迷宮搜秘。
(13) 太空怪侵略戰。
(14) 累加遊戲。
(15) 爲磁碟命名。
(16) 任意定義鍵盤。
(17) 捉蟲訓練問題。

7
(1) 會說話的蘋果
(2) 私人財政預算系統
(3) 私人收入會計系統
(4) 高解像字符繪劃器
(5) 圖像轉載器
(6) 螢幕文本編輯處理器
(7) 濃縮圖片程式修訂加強版本
(8) 蘋果杯大賽車
(9) 十六個難度的食鬼遊戲
(10) 高爾夫球遊戲。
(11) 空戰驚魂
(12) 美麗圖案設計家
(13) 更多色彩
(14) 列印的困擾
(15) 十一條空 SECTOR
(16) 捉蟲訓練
(17) 80字行咭例子程式

8
(一) APPLESOFT滙編器
(二) 最簡單的拆帶程式
(三) 英文打字練習
(四) 編印單個LABEL
(五) 高解像頁處理工具箱
(六) 簡易快速的記憶移動方法
(七) 眞正要動腦筋的小學四則運算
(八) 七海霸王 (DEPTH CHARGE)
(九) 原子黑洞
(十) 蛇之噩夢
(十一) 收銀機
(十二) 改良的閃動訊息
(十三) 兩欄索引顯示
(十四) 自動三角函數曲線繪劃系統
(十五) 任意鍵盤（三個）
(十六) 每期捉蟲問題
(十七) 奇妙的 &

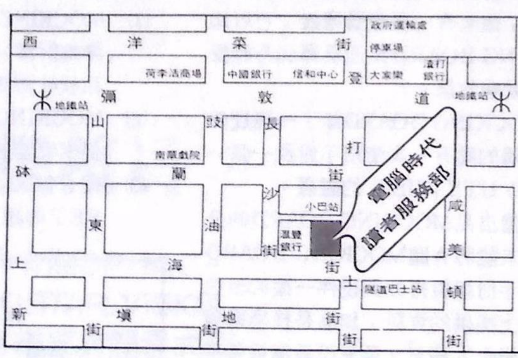
9
(1)磁碟檔案黃頁系統
(2)增強鍵盤輸入及螢幕輸出功能的系統
(3)將工具程式放入DOS佔用的三條磁軌上
(4)快速尋找及磁碟矩陣儲存與提取
(5)新DOS命令改換程式
(6)任意設定目錄索引顯示模式
(7)捉迷藏
(8)古堡之謎
(9)高空擲物
(10)有趣的排序
(11)捉虫問題及答案
(12)第八期捉虫問題答案
(13)印出高解像數目字

10
(1) 可隨意訂定格式之資料紀錄系統
(2) 太空戰士
(3) MARCO鍵盤
(4) 射鴨仔
(5) 奇妙的&——快速分類整理程式
(6) 40軌磁碟工具程式設計
(7) 變數表顯示
(8) 跳越加速器
(9) IQ方塊
(10) 快速檔案搜索
(11) 將DOS移上RAM咭
(12) 培養兒童的記憶能力
(13) 初中代數式運算
(14) 萬用數基轉換器
(15) 十六欄機械碼編印
(16) 捉蟲問題及答案

11
(1) 家庭購物備忘及存貨紀錄系統
(2) 可隨意訂定之資料紀錄系統（下）
(3) 奇妙的＆專欄——矩陣定位
(4) 40軌磁碟工具程式之(二) 單軌 INIT
(5) 兩個自動減省程式佔用之記憶空間應用程式
(6) 對我說
(7) 趣意盈盈的歡樂棒
(8) 英文語句文法分析練習系統
(9) 一行程式示例（共七個）
(10) 美化程式列表
(11) JACKIES HEAD
(12) 魔域迷宮
(13) 巨蟹求生記
(14) 自動日子記錄
(15) 空SEOCTOR圖表
(16) 捉虫訓練——FACTORIAL！

12
① 家庭開支及個人消費管理系統
② 電子郵箱
③ 40軌程式COPY工具
④ 諾曼第登陸戰（模擬戰爭思考遊戲）
⑤ 工兵掘地雷
⑥ 新改良蛇之噩夢
⑦ 魔域迷宮（APPLESOFT 版本）
⑧ 讓九個月大的嬰孩接觸電腦吧！
⑨ 有多項功能的AMPERSAND工具程式
⑩ 磁頭清潔器
⑪ 程式拯救者
⑫ TAB & XY
⑬ 符號逆輸型（RPN）計數機
⑭ 比RAM DISK更好的RAM咭控制中心
⑮ 圖形建造師
⑯ 捉虫訓練——SINE曲線
⑰ 一行程式

歡迎補購各期程式磁碟或雜誌



精選程式磁碟

編號：CA00A
內容：遊戲及教育程式
遊戲程式
1、捉球遊戲（第一期）
2、勇闖電腦室（第一期）
3、迷宮救美（第一期）
4、過四關（第一期）
5、沙蟹幻方局（第二期）
6、貨櫃處理（第二期）
7、賽跑遊戲（第三期）
8、步步追踪（第三期）
9、圍棋遊戲（第三期）
10、立體迷宮（第四期）
11、穿梭機哥倫比亞號（第四期）
教育程式
12、增強英文滙能力（第一期）
13、幼童數學四則運算（第一期）
14、數學套件程式（第二期）
15、動量不滅定律模擬法（第三期）
16、幼童認數字初階（第三期）
17、豐富設計的學前兒童教育程式（第四期）
18、應用數學套件程式之二（第四期）

編號：CA00B
內容：應用程式精選
1、電腦時代月曆（第一期）
2、COMPUTEXT 文字處理（第一期）
3、美術學繪劃器（第二期）
4、彩色電視／監視器色彩測試（第二期）
5、保留遊戲積分紀錄（第三期）
6、自製磁碟索引封套（第三期）
7、APPLE 編曲器（第三期）
8、雜誌目錄分類歸檔系統（第三期）
9、DOS WRITER（第四期）
10、電腦動畫製作原理（第四期）
11、將 RAM咭作爲磁碟機使用（第四期）

編號：CA00 C
內容：私人會計系統
(1) 私人財政預算 （第七期）
(2) 私人收入紀錄 （第七期）
(3) 私人會計程式 （第六期）

編號：CA00D
內容：工具程式集
(1) APPLESOFT程式偵錯系統（等六期）
(2) 將DOS 移上RAM 咭（第十期）
(3) MACRO 鍵盤 （第十期）
(4) RAM 咭控制中心 （第十二期）
(5) 程式編輯器 （第十三期）

每期程式磁碟現仍售29.9元；
精選程式磁碟每張售價45元
（除CA00C 只售30元）。

各期程式磁碟總目錄

13

(1) 行政思考決策輔助系統——成功人士必備工具
(2) 全能標貼製作器
(3) 大力水手——POPEYE
(4) 火樹銀花賀聖誕——平安夜
(5) 低解像圖輔助繪劇器
(6) 達爾文進化論的電腦模擬實踐
(7) 程式編輯器
(8) 磁碟檔案地圖
(9) 增加DOS的CATALOG暫停功能
(10) 新游標
(11) 40軌磁碟改動器
(12) 餅形圖
(13) 自己設信紙版頭
(14) 捉虫問題及答案
(15) 一行程式得獎作品

14A

(1) 高解像圖魔術師
(2) 以軟件來達成的音响合成器
(3) SCREEN SAVER
(4) 磁碟度身定造器
(5) DOS 的快速存取
(6) 一行程式得獎作品選
(7) HIDE：將程式隱藏
(8) 英文字義辨識練習
(9) 數恐龍學習加、減運算
(10) 宇宙城保衛戰（麥法基專欄）
(11) 最後一個立體打井遊戲
(12) 月球一小步，人類一大步
(13) 音量控制
(14) 將APPLESOFT程式轉爲TEXT檔案來編輯
(15) 捉虫訓練——DOS指令的執行

14B

DOS 3.3 FORMAT (SIDE A)
(A) 令蘋果發言的最佳工具MOCKING BOARD SPEECH WRITER
(B) 本港郵費計算諮詢系統
CP／M FORMAT (SIDE B)
(C) CP／M的密碼檔案
(D) 程式設計與FORTRAN所帶來的好處（雙精度計算例子）

15

(1) 活動電子訊息顯示板
(2) 借貸分期還款計算系統
(3) 麥法基遊戲專欄——DANGER RANGER
(4)「扁」臨香江
(5) 將高解像頁倒轉
(6) 機械碼輸入編輯器
(7) 程序分解擴展器
(8) 利用印字機印出多款字形
(9) 檔案密碼鎖
(10) 兩個新DOS指令之一MREAD
(11) 電腦繪圖專欄——立體繪圖的原理示範程式
(12) 氫氧碳化學分子排列結構模擬
(13) 豬仔錢箱
(14) 3－6－9鬥智遊戲
(15) 超級越軌大賽
(16) 電腦音樂動畫——MONICA
(17) 一行程式入選作品
(18) 捉虫問題及答案

16

(1) 專業室內設計師
(2) 大忙人時間顧問
(3) 學習迷宮產生的原理及自編迷宮遊戲
(4) 使用更方便的BOOT DRIVE 2 軟件
(5) LIST大師
(6) CATALOG HELPER
(7) 40軌磁碟SECTOR使用情況顯示
(8) 磁碟保護器——COPY VTOC法
(9) 兩個新DOS指令MREAD & MWRT（下）
(10) 爲你的程式配上背景音樂
(11) 成雙成對
(12) 吃不到的葡萄不是酸的!!
(13) 動態／策略兼備的戰爭遊戲：BEACH HEAD
(14) 一行程式入選作品
(15) 捉虫問題及答案
(16) 組合線性方程式解答系統
(17) 太空侵略者封閉戰CLOSING－ALIEN

補購請到——
上海街438號同珍商業中心11字樓1104室（上海街／登打士街交界）
電話：3-312007

COMPUHOUSE

歡迎大家到來讀者服務部參觀選購

代理經銷：48 K CPU　16 K RAM 咭　Z 80咭　各類高級磁碟
128 K RAM 咭　80 COL 咭　IBM兼容電腦　PRINTER咭

最新產品：

★CP/M FORMAT 40 (VERSION II)
將CP／M格式磁碟由35軌改成爲40軌

★GPLE使用方法詳析（即將出版）
BEAGLE BROTHER 廠出品的一個強效全域編輯器，我們深入剖析其內在功能並將使用方法編寫成書。

★各種磁帶保護破解天書（即將出版）
★MOCKING BOARD技術應用(即將出版)
★TAKE 1 電影大獎優勝作品選（磁碟版本）

★CHECK SUM軟件

■破解答案（磁碟版本，兩面）
(經已出版)
INSTITUE 及 LUCIFER'S RELAM
豪門恩怨/TRACER SANCTION
每張售20元

徵求一行程式有獎遊戲

在上月我們收到許多〔一行程式〕的投稿，其中不乏精彩的作品。

這個遊戲旨在鼓勵大家動腦筋，利用最簡短的程式，設計一些有創意的效果。

今次未刊登你的作品的讀者請勿灰心，可能我們會在下一期登出。

各位投稿者請在來稿中加上程式內容說明。以前各期的得獎者請盡快前來讀者服務部領取應得的獎品。

各位讀者寫來程式時，必須用多行的形式寫來（參考本文的各位得獎例子），而且要書寫清楚。投稿請寄來CPO BOX 71193 。入選者可獲得任何一期的程式磁碟一張（請註明要的磁碟）。來稿信封請勿封口。 ■

本程式可將 1 至3999之間的任何數字轉變爲羅馬數字，只需RUN程式，按入 1 至3999間的任何一數，電腦便會顯示該數的羅馬數字。沈志明君可取得第七期程式磁碟一張

```
10    INPUT "(0-3999)->";N%
    : N$ = STR$ (N%)
    : L = LEN (N$)
    : FOR D = 1 TO L
    : M = VAL ( MID$ (N$,L - D + 1,1))
    : READ I$
    : A$ = MID$ (I$,((M < 4) + (M = 9))
      * 3 + (M > 4) * 2 + (M = 4),(M
      > 0) * M - ((M = 4) + (M = 9))
      * 2 - (M > 4) * 4)
    : R$ = A$ + R$
    : NEXT
    : PRINT R$
    : RUN
    : DATA "IVIIIX","XLXXXC","CDCCCM","
      ??MMM?"
```

本程式以造型表在高解像中繪畫出一隻貓頭鷹的頭。整個圖是由線的旋轉及放大所繪成的。

黃偉樑君可取得第 8 期磁碟一張。

```
10    FOR I = 16384 TO 16390
    : READ D
    : POKE I,D
    : NEXT
    : HOME
    : HGR
    : HCOLOR= 3
    : POKE 232,0
    : POKE 233,64
    : SCALE= 50
    : FOR I = 1 TO 128
    : ROT= I
    : XDRAW 1 AT 70 + I,80
    : NEXT
    : SCALE= 10
    : FOR I = - 3 TO 3
    : ROT= 64 + I
    : XDRAW 1 AT 134 + I,130
    : NEXT
    : DATA 1,0,4,0,4,128,0
```

本程式是用來繪製BAR CHART 之用。

鄭家輝所要的第18期磁碟要待下期出版後才可以得到。

```
40    INPUT "HOW MANY ITEMS";N
    : INPUT "BOUNDARY:";B
    : FOR I = 0 TO N - 1
    : PRINT "ITEM    :";I + 1;
    : INPUT A(I)
    : NEXT
    : HGR2
    : HCOLOR= 3
    : HPLOT 0,0 TO 0,179
    : HPLOT TO 279,179
    : FOR I = 0 TO N - 1
    : FOR T = 0 TO 3
    : HPLOT I * 5 + T,179 - A(I) / B
      * 179 TO I * 5 + T,0
    : NEXT T
    : NEXT I
```

本程式能在畫面顯示磁碟上已用磁區的分配圖和計算空置磁區的數目。S. H. WONG 君可取得第十四期磁碟一張。

```
10    HOME
    : A = 46067
    : U$(0) = ":"
    : U$(1) = "*"
    : CALL 45047
    : FOR T = 0 TO PEEK (46063) - 1
    : B = PEEK (A) * 256 + PEEK
      (A + 1)
    : FOR S = 0 TO 15
    : D = B / 2
    : E = INT (D)
    : U = U + (D < > E)
    : HTAB T + 1
    : VTAB S + 2
    : PRINT U$(D = E)
    : B = E
    : NEXT
    : A = A + 4
    : HTAB T + 1
    : VTAB 19
    : PRINT C
    : C = (C + 1) * (C < 9)
    : NEXT
    : VTAB 22
    : PRINT U;"FREE SECTOR"
```

電腦時代讀者服務部與 COMPUHOUSE 聯合主辦

APPLE FAN CLUB

APPLE FAN CLUB（以下簡稱AFC）屬於一個不牟利性質的組織，旨在聯繫蘋果電腦用家，促進大家對電腦的應用。歡迎蘋果電腦用家參加。

AFC 提供的權利包括：

① 免費借用最新出版的蘋果電腦軟件。
② 以批發價錢供應會員各類磁碟、磁碟機、週邊咭。
③ 獲得每月三份的十日刊通訊錄，通訊錄報導各方面的最新軟件出版消息，完全免費。
④ 爲讀者搜羅各類的MANUAL 。
⑤ 供給各種新軟件的BACK UP或破解方法。
⑥ 會員可以安坐家中足不出戶取得所借用的軟件。

每個月，港九的高登及國都商場都推出有不少的新軟件，包括遊戲及工具軟件。我們將一一的搜羅這些軟件，存放在AFC內的SOFTWARE LIBRARY 內，以便供會員借用。每隔十天我們會寄出一份最新軟件出版的消息動態及名單，會員收到通訊錄後，可隨意選擇要借用的軟件，填妥隨附的借用申請表格連同空磁碟（我們亦以批發價錢供應各類型磁碟）寄來AFC。我們收到借用表格後，在五個工作天內便會將會員要借用的軟件寄出。或者申請者可於指定時間到電腦時代讀者服務部攞取。每名會員每月可免費借用六個軟件，若要多借軟件，每個只收取象徵式一元的服務費（可用郵票代付），此外，別無其他收費。同時，我們亦提供有關的軟件的MANUAL ，我們只收回影印成本。

招收基本會員（名額有限　報名從速）

參加AFC資格：只要是蘋果電腦用家即可。

收費辦法：參加者每年須繳付年費180元。亦可分期繳交，辦法如下：

(1) 一次過繳交一年會費　$180
(2) 分二期每半年繳交一次 $108

各位只要填妥下列表格，連同劃線支票寄來九龍中央郵政信箱CPO BOX 71193，電腦時代出版社收卽可，支票抬頭填寫〔COMPUTING AGE PUBLISHER〕，信封面請註明〔AFC〕。分期繳付會費的申請先繳付首期會費。

APPLE FAN CLUB 申請入會表格

申請者姓名：(英文) _____ (中文) _____

通訊地址：(英文) _____

_____

(中文) _____

_____

通訊電話：_____ 職　業：_____

支票號碼及銀行：_____ 申請日期 _____

特惠優待證號碼＊ _____ 電腦時代訂閱戶＊ _____ COMPUSOFT磁碟月刊訂閱戶＊ _____

＊假如是訂閱戶請填寫訂戶號碼

繳付會費辦法。

☐ 一次繳付　☐ 分二期繳付

NO. | | | | | | | | |
Official Use

＊＊香港以外地區申請者需加附郵費（空郵）40元，另外每次申請借用軟件需自付郵費。
請容許十五至三十日時間辦理登記會員手續。我們會發信通知。

電腦時代程式磁碟

每張只售HK$29.90

電腦時代每期都刊登多個精彩的程式，其中必然有你喜歡的。當你讀完文章興緻勃勃的坐下來欲進一步深入了解程式的精彩處時，看見那大堆冗長的列表，就甚麼興趣也沒有了！那麼，你爲何不購買一張「電腦時代程式磁碟」呢！

每期出版的電腦時代所登載的程式，均全部紀錄入一張高品質的磁碟上，並隨雜誌同時出版。爲省卻打入程式的麻煩，徒然浪費你寶貴的時間，免除了花精神捉錯打字的痛苦，今日就請就近的特約經銷處或電腦時代讀者服務部/COMPUHOUSE購備一張。

程式磁碟每張只售港幣 29.90 元。

第17期程式磁碟內容

*All Programs are in Apple version

(1) 日文書信文字處理系統
(2) APPLESOFT 全域編輯器
(3) II PLUS 亦可擁有視窗顯示功能
(4) 仿效MOUSE PAINT設計的AP PAINT
(5) 商業發貨單
(6) 把損壞的磁軌鎖住
(7) 配合單軌INIT使用的INIT 40軌
(8) WILDCARD解拆保護原理透視
(9) 兒童串字
(10) 英文聽力測驗
(11) 金字塔數字堆
(12) 麥法基遊戲專欄：TIME PILOT
(13) 武士忠魂
(14) MUSIC CONSTRUCTION SET背景音樂問題
(15) GRAPHIC SOLUTION 動畫DEMO
(16) VISICALC 篇——DEMO
(17) 一行程式獲選作品
(18) 第十七期捉虫訓練

本期程式磁碟雙面灌錄　讀者留意

爲方便各區讀者購買程式磁碟，我們新增加多個特約經銷處：

以下爲各特約經銷處地址：

中環
百老滙攝影器材有限公司（第五店）
城多利皇后街3號（中環街市側）
電話：5－235513

亞當影音
士丹利街 28 A（陸羽茶樓側）
電話：5－240723

三聯書店讀者服務中心
香港中環城多利皇后街九號（中環街市側）
電話：5~250102

銅鑼灣
百老滙攝影器材有限公司（第十四店）
軒尼斯道529號地下（三越百貨公司對面）
電話：5－770181

太古城
百老滙攝影器材有限公司（第七店）
太古城中心第232店
電話：5－672006

西環
創業雜誌中心
西環屈地街22號地下　明愛中心對面
電話：5-478338

旺角
百老滙攝影器材有限公司　（第1,3,9,10店）
彌敦道758號在下（凱聲戲院側）
電話：3－816931
彌敦道700號旺角中心第二期地下
電話：3－943827
豉油街26號地下（近商務書局）
電話：3－884331
亞皆老街 60 B 地下（世界書局對面）
電話：3－956367

深水步
百老滙攝影器材有限公司（第十三店）
青山道148號地下（華聲戲院對面）
電話：3－7295938

尖沙咀
智源書局
金巴利道 27－33號永利大厦2字樓A座
電話：3 － 678482 3－678414

荔枝角
百老滙攝影器材有限公司（第二店）
美孚新邨第四期百老滙街 79號（影都戲院側）
電話：3－7424923

觀塘
百老滙攝影器材有限公司（第四、十二店）
物華街3號地下（皇上皇側）
電話：3－419513
同仁街 35 號地下（大家樂側）
電話：3－449940

荃灣
百老滙攝影器材有限公司（第十一店）
沙咀道 230 A（戴麟趾夫人健康院對面）
電話： 0-4938926

屯門
恒達時電腦公司
屯門華都花園地下商場8號室
電話：0-252577

新增設特約經銷商：
教育專業人員協會
（佐頓及北角會所）

電腦時代讀者服務部/ CompuHouse

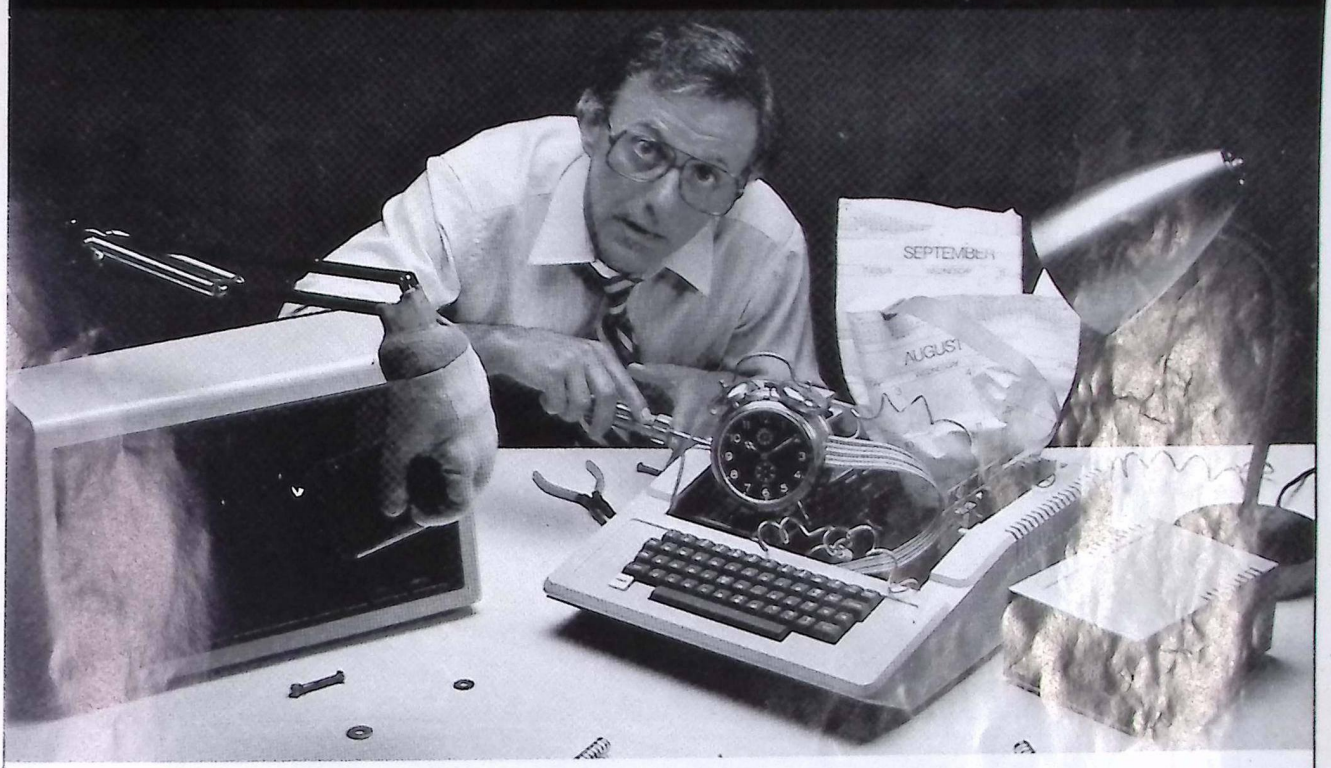
地址：旺角上海街438號同珍商業中心11字樓1104室
（上海街／登打士街交界）　電話：3-312007

補購各期程式磁碟可前往讀者服務部／COMPU HOUSE

Apple is the trademark of Apple Computer Inc.

何必自己動手

免費 由專業人員上門爲你修理微型電腦



請即加入本公司之保養及免費維修服務

24小時熱綫電話服務：3-848129

微型電腦保養維修中心

ROM & RAM computer ltd

SHOP: 582–592, Nathan Road, Basement Shop No. B35 Sino Centre, Kowloon. Tel: 3-306839

OFFICE: Rm. 1905 Witty Coml. Bldg., 1A–1J, Tung Choi St., Kowloon. TEL: 3-848129

Telex: 57551 ROMHX

隨書附送價值超過150元的購買程式磁碟優待券

精裝珍藏本，顧名思義，是將過去出版的電腦時代第1至12期所有的精彩內容滙集成一巨冊，其中包括超過一百個的軟件程式使用說明，數十篇有關各類週邊硬件深入解剖報告、熱門軟件評介及用法文章。是所有擁有APPLE或APPLE兼容電腦的用家必須珍藏的巨獻。

可能你已擁有第一至十二期，但這冊精裝珍藏本依然有你值得擁有的價值，因為我們在這冊合訂本中加插了多篇從未刊登過的文章及程式。而且我們更隨書附送價值超過150元的購買程式磁碟優待券。

電腦時代精裝珍藏本全書厚度超過200餘頁，採用穩固的膠背釘裝。每冊僅售50元。

電腦時代精裝珍藏本

(1-12期合訂本)

5月出版　各大報攤有售

學校電腦輔助教育

50 ■英文聽力測驗

兒童電腦輔助教育

66 ■兒童串字訓練

最新遊戲軟件評介

79 WARRIORS TRACER SANCTION
SHADOWKEEP 魯賓遜飄流記
GEMSTONE WARRIORS 豪門恩怨
WINDOW CLASS F-15

BACK UP資料館

82 ■WILDCARD 能夠解拆保護的原理透視
83 抽不死的啤酒佬
83 孫悟空任你玩
83 打不死的RUN FOR IT
78 有保護遊戲之拆圖方法

精品店

21 PDL有趣的用途
74 程式開首LIST的特殊效果
45 妙趣CATALOG

讀者園地

88 ■一行程式參加得獎者
81 ■捉蟲訓練
86 各期程式磁碟總目錄
46 讀者意見調查
85 電腦時代技術叢書出版消息
43 COMPUSOFT磁碟月刊廣告介紹

DOSF

特稿

四大磁碟系統磁碟資料存放格式詳析(第一篇DOS3.3) 15
揭開老鼠仔神秘的一面 20
■ 仿效MOUSEPAINT而設計的APPLEPAINT 22
解決DOS 3.3 SUB-DIRECTORY的方法 14
■ MUSIC CONSTRUCTION SET背景音樂問題 63

高級篇

dBASE篇：認識dBASE II的第一步 28
■ VISICALC篇：處理超越函數及產生圖表的研究 31

應用軟件庫

■ 日文文字書信處理系統 33
■ APPLESOFT全域編輯器 35

中小型工商業應用

■ 商業發貨單 12

小型製作專欄

價廉物美的JOYSTICK AUTOFIRE製作 44
如何調節光暗的80字行咭 69

軟件縱橫談

■ 一套嶄新的動畫製作軟件——GRAPHIC SOLUTION 48

專欄

機械語言入門：如何控制磁碟機馬達活動 56
PRODOS縱橫：建立一個更好的DOS系統 58
麥法基專欄：掀開蘋果高解像畫面的神秘面紗(二) 60

蘋果工具箱

■ II PLUS亦能擁有視窗顯示 51
■ 配合SINGLE TRACK INIT使用的INIT 40 TRACK 53
■ TRACK LOCKER把損壞的磁軌鎖住 55

齊齊樂開心坊

■ GRAPHIC MAGICIAN編寫的遊戲TIME PILOT 70
■ 武士忠魂 72
■ 金字塔數字堆 75

電腦時代

NO.17

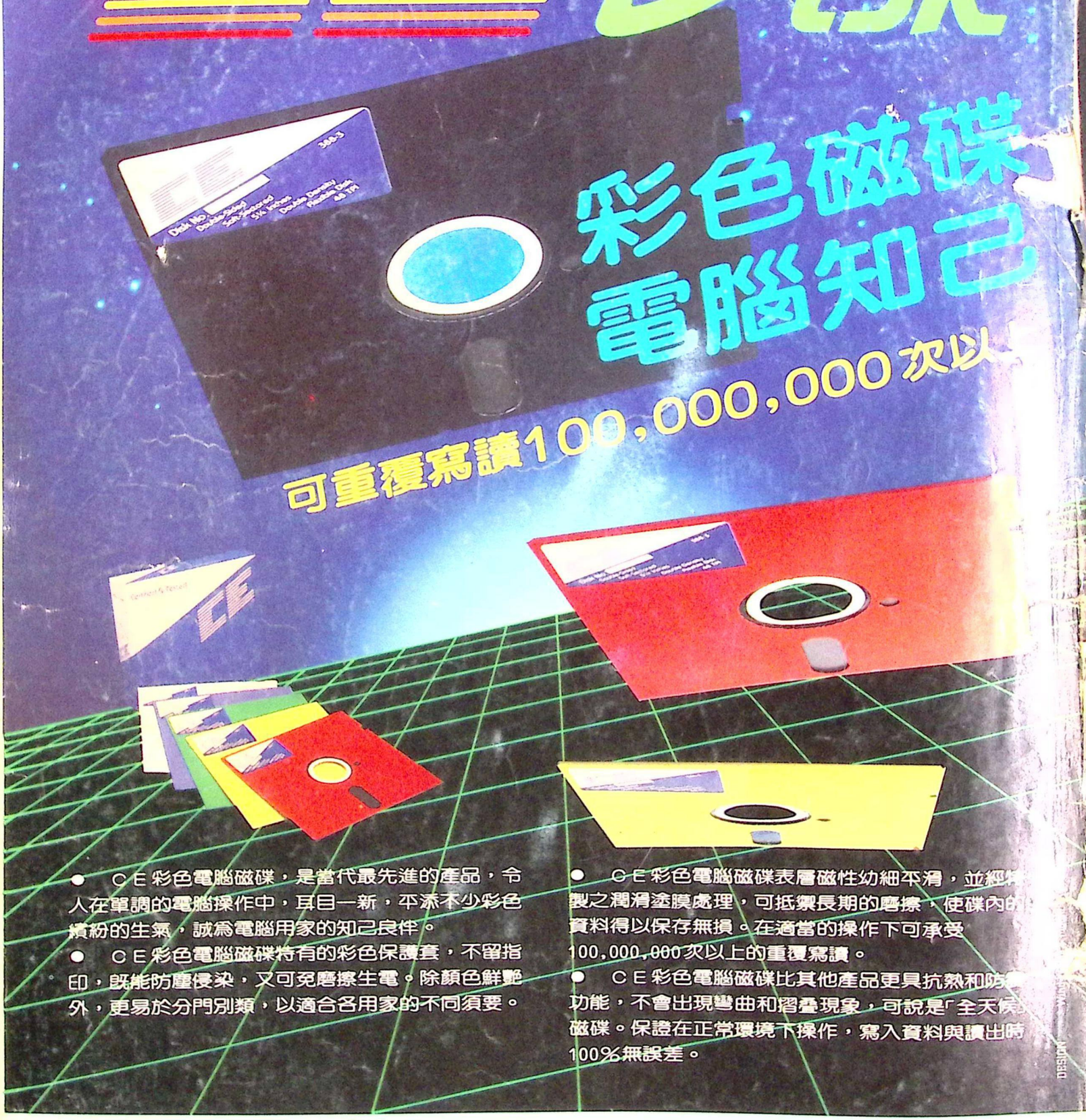
本刊每月廿五號出版。一年十二期。全年訂閱費120港元，雜誌連程式磁碟爲475港元。海外訂戶請參看訂閱收費表。

◘出版：電腦時代出版社 ◘督印人：伍小華 ◘編輯：電腦時代編輯部 ◘軟件支援：COMPUHOUSE ◘通訊地址：九龍中央郵箱71193號 ◘通訊電話：3-312007 ◘讀者服務部地址：旺角上海街438號同珍商業中心11字樓1104室（登打士街交界）
◘承印者：永恒印刷公司 ◘發行人：同德書報社

CE Disk

彩色磁碟 電腦知己

可重覆寫讀100,000,000次以



● ＣＥ彩色電腦磁碟，是當代最先進的產品，令人在單調的電腦操作中，耳目一新，平添不少彩色繽紛的生氣，誠為電腦用家的知己良伴。

● ＣＥ彩色電腦磁碟特有的彩色保護套，不留指印，既能防塵侵染，又可免磨擦生電。除顏色鮮艷外，更易於分門別類，以適合各用家的不同須要。

● ＣＥ彩色電腦磁碟表層磁性幼細平滑，並經特製之潤滑塗膜處理，可抵禦長期的磨擦，使碟內的資料得以保存無損。在適當的操作下可承受100,000,000次以上的重覆寫讀。

● ＣＥ彩色電腦磁碟比其他產品更具抗熱和防[illegible]功能，不會出現彎曲和摺疊現象，可說是「全天候」磁碟。保證在正常環境下操作，寫入資料與讀出時100%無誤差。

香港代理：華益企業公司
NO.31-33 KING'S ROAD, 1ST FLOOR,
(NEAR LAU LI ST.) NORTH POINT.
G.P.O. BOX 10617,
HONG KONG

TELEX NO: 75632 PECO HX
CABLE ADDRESS: "PETCOMY" HONG KONG
TELEPHONE: 5-784928, 5-783129
BANKER: HANG LUNG BANK LTD.